VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

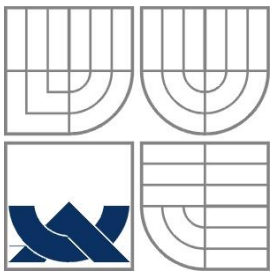
ZÍSKÁVÁNÍ ZNALOSTÍ Z WEBOVÝCH LOGŮ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

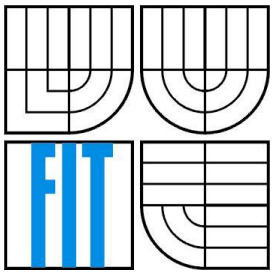
AUTOR PRÁCE
AUTHOR

Bc. Vladimír Vlk

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ZÍSKÁVÁNÍ ZNALOSTÍ Z WEBOVÝCH LOGŮ

KNOWLEDGE DISCOVERY FROM WEB LOGS

DIPLOMOVÁ PRÁCE
MASTER‘S THESIS

AUTOR PRÁCE — Bc. Vladimír Vlk
AUTHOR

VEDOUCÍ PRÁCE — Ing. Vladimír Bartík, Ph.D.
SUPERVISOR

BRNO 2013

Abstrakt

Tato diplomová práce se zabývá tvorbou aplikace provádějící předzpracování dat z webového logu a nalezení asociačních pravidel. První část se zabývá pojmem získávání znalostí z Webu. Druhá část je věnována získávání znalostí z užití Webu a souvisejícím pojmům, jako předzpracování dat, nalezení a analýza vzorů, atd. Třetí část se zabývá návrhem aplikace. Čtvrtá část je věnována popisu implementace aplikace. Poslední část se zabývá experimenty s aplikací a interpretací výsledků.

Abstract

This master's thesis deals with creating of an application, goal of which is to perform data preprocessing of web logs and finding association rules in them. The first part deals with the concept of Web mining. The second part is devoted to Web usage mining and notions related to it. The third part deals with design of the application. The forth section is devoted to describing the implementation of the application. The last section deals with experimentation with the application and results interpretation.

Klíčová slova

Dolování dat, Web log, předzpracování, dolování dat z užití, dolováni dat z Webu, asociační pravidla, Apriori

Keywords

Data mining, Web log, preprocessing, web usage mining, web mining, association rules, Apriori

Citace

Vladimír Vlk: Získávání znalostí z webových logů, diplomová práce, Brno, FIT VUT v Brně, 2013

ZÍSKÁVÁNÍ ZNALOSTÍ Z WEBOVÝCH LOGŮ

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Vladimíra Bartíka, Ph.D.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Vladimír Vlk
22.5.2013

Poděkování

Na tomto místě bych rád poděkoval vedoucímu mé diplomové práce Ing. Vladimíru Bartíkovi, Ph.D. za čas, který mi věnoval, za trpělivost, pomoc a rady, které mi poskytl při vytváření této práce.

Obsah

1 Úvod

Síť internetových stránek (World Wide Web, WWW) se neustále zvětšuje a vyžaduje stále více prostoru na webových serverech. S růstem sítě internetových stránek se zvětšuje i množství dokumentů uložených v elektronické podobě na Webu. Tato skutečnost přispívá ke zvýšení složitosti návrhu webových stránek. Z tohoto důvodu se začala rozšiřovat analýza způsobu používání webových stránek, jejímž cílem je usnadnit práci designérům stránek, stejně jako přizpůsobit webovou stránku potřebám a chování uživatelů. Analýza také může vézt ke zjištění řady dalších užitečných informací. Tyto informace se pak často využívají ke komerčním účelům. Stále větší množství firem, které se prezentují na Webu, se zajímá o to, jakým způsobem je možné efektivně oslovit návštěvníky stránek, zjistit proč naše stránky navštěvují, atd. A právě analýza používání webových stránek je možností jak získat odpovědí na tyto otázky. Získané odpovědi nemusí být vždy přesné, ale jejich přesnost je dostatečně vysoká, aby mohla být použita pro závažné rozhodnutí.

Cílem této práce je vytvořit nástroj řadící se do oblasti získávání znalostí z užití Webu a provádějící předzpracování dat z webového logu a nalezení asociačních pravidel. Nalezená asociační pravidla budou sloužit pro optimalizaci webové stránky, jejíž log soubor byl zkoumán. Nástroj bude pracovat s formátem log souboru typu NCSA Combined, který je pro danou úlohu vhodný.

V kapitole číslo 2 se práce zabývá obecnou definicí pojmu získávání znalostí z Webu, jednotlivými kroky v procesu získávání znalostí z webu a v poslední části oblastmi, ze kterých je možné informace získávat. Následující kapitola je věnována upřesnění pojmu získávání znalostí z užití Webu. Dále se v rámci této kapitoly věnuje práce jednotlivým krokům procesu získávání znalostí. Důraz je kladen na proces předzpracování dat, asociační pravidla a jejich získávání. Závěr kapitoly je věnován formátům souborů webových logů. Kapitola č. 4 se zabývá návrhem aplikace, jejíž vytvoření je cílem této práce. V další kapitole se práce zabývá popisem způsobu implementace aplikace. Postupně je uveden způsob implementace jednotlivých kroků, uvedených v rámci kapitoly 3, v aplikaci. Jsou představeny doplňující funkce aplikace, které nejsou přímou součástí získávání znalostí. V kapitole je také popsán způsob použití aplikace a účel jednotlivých nastavení. Předposlední kapitola se věnuje interpretaci získaných výsledků a vlivu jednotlivých nastavení na získání asociačních pravidel. Předposlední část kapitoly je poté věnována rozboru časové složitosti vyhledání asociačních pravidel. Poslední kapitolou je závěr, který obsahuje shrnutí práce a navrhovaná rozšíření aplikace do budoucnosti.

2 Získávání znalostí z webu

Získávání znalostí z Webu (angl. Web mining, nebo také data mining the Web) patří vzhledem k velkému množství informací dostupných na webu mezi důležité činnosti v oblasti dolování dat. V oblasti získávání znalostí z webu jsou nové informace získávány pomocí extrakce potenciálně užitečných informací dostupných na Webu. Nově nabyté znalosti jsou používány např. ke zkvalitnění vyhledávání relevantních informací pomocí vyhledávácích serverů, marketingovým účelům, přizpůsobení stránek požadavkům uživatelů, atd. Jak již bylo uvedeno, získávání znalostí z Webu patří do oblasti dolování dat a samotné získávání znalostí je založeno na použití technik z oblasti dolování znalostí. Proces získávání znalostí lze na základě KDD (Knowledge discovery in databases) rozdělit na pět fází. Pojem KDD v odborné literatuře poprvé definoval Fayyad a kol. v roce 1996 [1]. Na základě této definice je KDD netriviální proces získávání implicitních, předtím neznámých a potenciálně užitečných informací.

Pět základních fází KDD jsou následovné:

1. Výběr (selection)
2. Fáze předzpracování (preprocessing)
3. Transformace (transformation)
4. Dolování (data mining)
5. Vyhodnocení/interpretace (evaluation/intepretation)

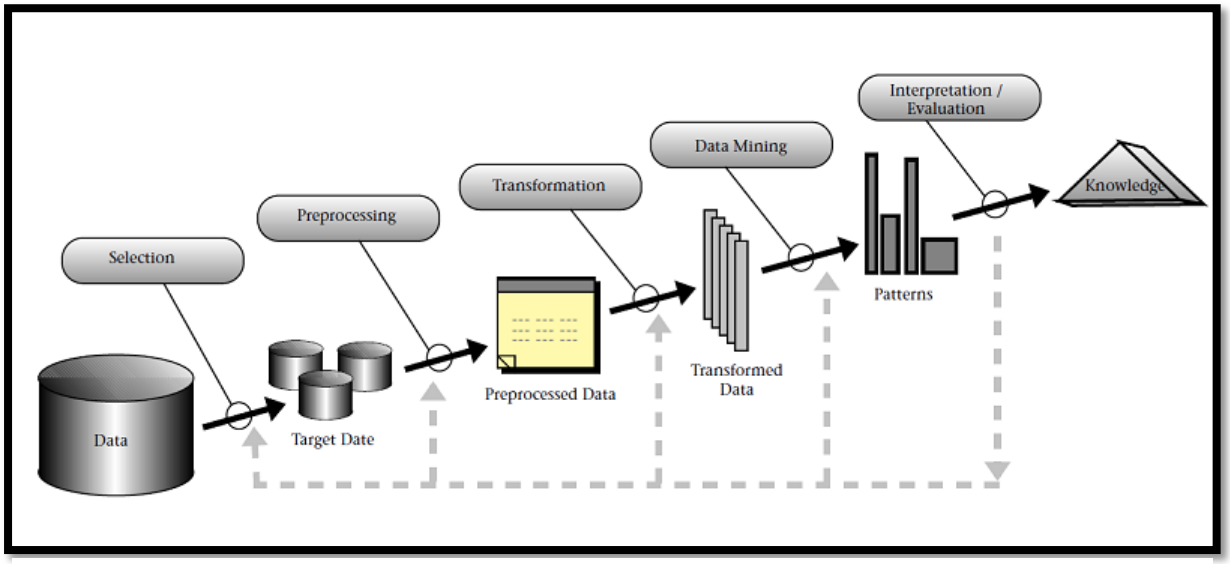
1. **Výběr** – hlavním úkolem této fáze je získání informací, které budou použity pro dolování znalostí. Důležité je z velké množiny dat vybrat pouze tu část, která může obsahovat užitečná data pro dolování. V oblasti získávání informací z webu se podle zdroje informací dělí tato oblast do tří kategorií. Tyto kategorie budou blíže interpretovány později.

2. **Fáze předzpracování** – tato fáze se zaměřuje především na vyčištění a zkompletování informací získaných v předešlé fázi. Vyčištěním rozumíme odstranění nerelevantních dat, šumu a nesprávných dat. Zkompletováním informace chápeme doplnění chybějících atributů. V této fázi můžeme také vynechat odlehlé hodnoty, tedy hodnoty, které jsou příliš vzdáleny od ostatních a mohly by negativně ovlivnit výsledné znalosti.

3. **Transformace** – úkolem této fáze je transformovat výstup fáze pro předzpracování do formátu vhodného pro fázi dolování (data mining).

4. **Dolování** – představuje klíčovou část celého KDD procesu. V této části je na data získaná z předchozí části aplikován algoritmus sloužící pro generalizaci dat, tedy přesněji pro automatické nalezení vzorů. Použitým algoritmem může být např. algoritmus pro shlukování (clustering), klasifikaci (classification), regresi, nalezení asociačních pravidel, atd.

5. **Vyhodnocení/intepretace** – poslední fáze se zabývá validací a intepretací získaných vzorů.



Obrázek 2-1 : Proces KDD (převzato z [1])

Data pro získávání znalostí z webu můžeme získat z různých zdrojů, např. ze serveru, od klienta, z firemní databáze, atd. Data se liší nejen svým původem, ale také klasifikací. Rozlišujeme tři kategorie pro získávání znalostí z Webu:

Získávání znalostí z obsahu

Zabývá se dolováním znalostí z obsahu webových dokumentů. Obsahem webových dokumentů je myšleno velké množství různých dokumentů, jako např. text, zvuk, video, hypertext, obraz, atd. Oproti tradičním technikám text miningu tedy umí pracovat i s částečně strukturovanými daty. Obsahem webové stránky jsou tedy nestrukturovaná data ve formě textu, částečně strukturovaná data v podobě HTML dokumentu a strukturovaná data jsou reprezentována např. daty v tabulkách. Získávání znalostí z obsahu (Web content mining) je z velké části využíváno pro hodnocení relevantnosti obsahu na uživatelem zadaný vyhledávací dotaz. Tato metoda tedy provádí shromáždění a roztřídění velkého množství informací dostupných na webových stránkách a jejím hlavním cílem je poskytnout uživateli co nejlepší možnou informaci na jím zadaný dotaz. Jde o nezbytný nástroj pro prohledání obrovského množství HTML dokumentů, obrázků a textu na webových stránkách.

Získávání znalostí ze struktury

Hlavním cílem získávání znalostí z Webu je zabývat se strukturou odkazů v rámci Webu. Získávání znalostí ze struktury využívá doplňkové informace, která je obsažena ve struktuře hypertextu. Důležitým úkolem je tedy identifikovat relativní důležitost stránek, které se sice při analyzování zobrazují stejně, ale ve skutečnosti jde o odlišné stránky. Proto je snahou nalézt model představující strukturu odkazů na Webu. Takový model je založen na typologii odkazů a může být využíván pro kategorizaci webových stránek. Je užitečný také např. pro definování vztahu mezi různými webovými stránkami. Pro modelování topologie Webu jsou využívány algoritmy, jako HITS, PageRank, atd. Uvedené metody slouží k výpočtu relevance webové stránky.[2]

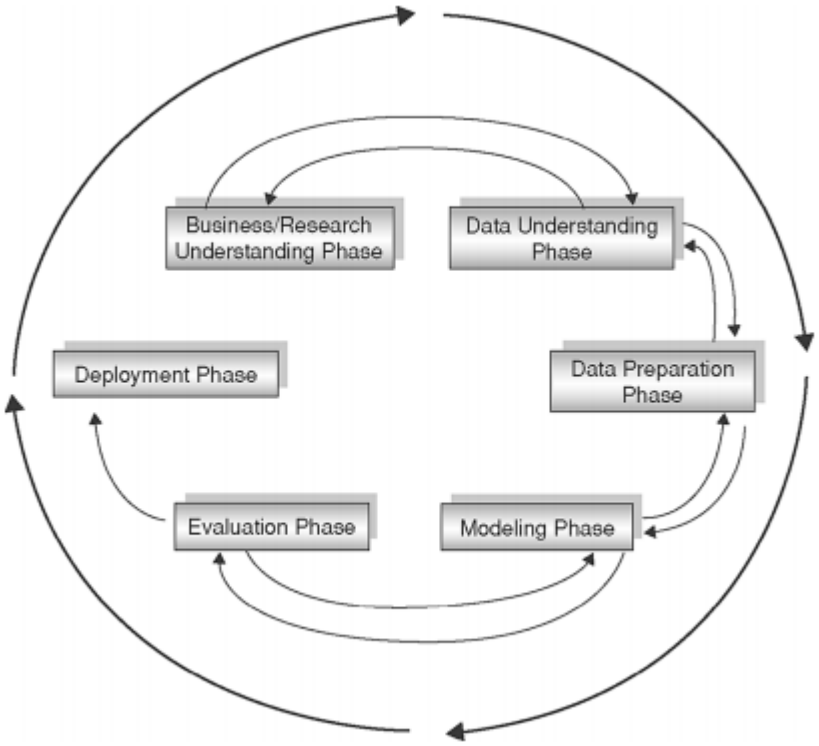
Získávání znalostí z užití

Jelikož je tato práce zaměřená především na kategorii získávání znalostí z užití, bude daná kategorie rozebrána podrobněji v následující samostatné kapitole. Zaměříme se postupně na všechny fáze získávání znalostí z užití a také na využití získaných informací. Nejvíce se však budeme věnovat fázi předzpracování dat.

3 Získávání znalostí z užití

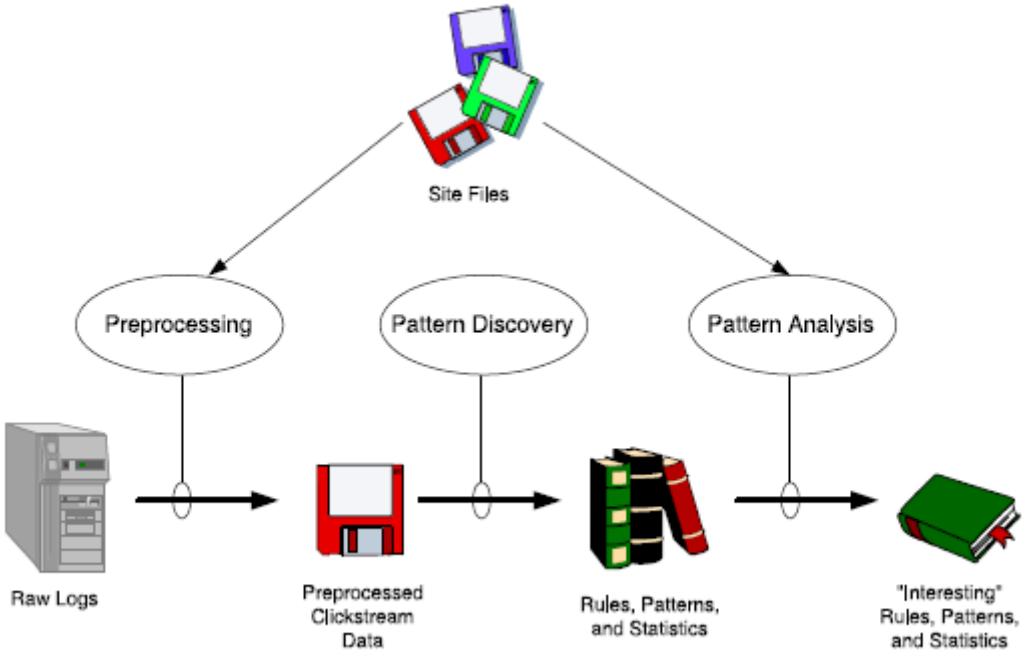
Získávání znalostí z užití je aplikace technik pro dolování dat, jejímž cílem je odhalit a porozumět způsobu užití webových stránek z webových dat [3]. Získané znalosti jsou následně použity pro upravení webových aplikací tak, aby lépe vyhovovali potřebám uživatelů. Získávání znalostí z užití se tak oproti předchozím dvěma kategoriím zabývá chováním uživatelů na Internetu. Z tohoto důvodu je získávání znalostí z užití hojně využíváno pro marketingové účely a pro úpravu webové stránky uživateli na míru, tzn., jde o personalizaci stránek.

Stejně jako ostatní techniky dolování dat definuje i technika získávání znalostí z užití kroky, jejichž postupné provedení vede k získání požadovaných znalostí. Stejně jako u klasického dolování dat, mohou být tyto kroky rozděleny pomocí CRISP-DM (Cross Industry Standard Process for Data Mining), který zobrazuje obrázek 3-1.



Obrázek 3-1 CRISP-DM převzato z [4]

V této práci však použijeme rozdělení podle obrázku 3-2. Jak můžeme vidět, jde o rozdělení na čtyři fáze. První fázi, která nemusí být z obrázku zcela patrná, je fáze nalezení zdroje dat. Dalšími fázemi jsou pak předzpracování, nalezení vzorů a analýza. Všechny zmíněné fáze budou v následujícím textu postupně rozebrány.



Obrázek 3-2 Proces Získávání znalostí z užití (převzato z [3])

3.1 Nalezení zdroje dat

Pro získávání znalostí z užití existuje více možných zdrojů dat. Vhodnost použití a výhody jednotlivých zdrojů se liší, a proto ve fázi nalezení zdroje rozhodujeme, který zdroj bude pro danou úlohu nejvhodnější. Tedy např. data ze serveru budou vhodnější pro získání informací z jedné webové stránky, naopak data od klienta se lépe hodí pro zkoumání činnosti uživatele na internetu. Jak již bylo naznačeno, dva základní zdroje dat jsou klient a server. Na tyto zdroje dat se podíváme blíže.

Data ze serveru

Velkou výhodou dat ze severu je jejich dostupnost. Tento typ dat je dostupný vždy, protože každý webový server ukládá logovací soubory. A právě webové logy jsou nejčastějším zdrojem dat pro proces získávání znalostí z užití. Informace uložené ve weblogu se mohou u různých serverů lišit. Jaké informace budou uložené, závisí na konfiguraci serveru. Jedním z nejčastějších formátů je Common log format, který obsahuje IP adresu nebo doménová jméno uživatele, uživatelské jméno, datum a čas, požadovanou adresu URI (uniform resource identifier), server status a počet přenesených bytů. Dalším možným formátem je tzv. Extended common log format, který přidává k předchozímu formátu další dvě pole. Těmito poli jsou user agent poskytující informaci o prohlížeči klienta, atd. a pole referrer, které udává URL předchozí stránky navštívené uživatelem. Kromě těchto formátů existují i další, např. Microsoft IIS log formát. Pro představuju, jak může logovací soubor vypadat, však stačí výše uvedené formáty.

Z výše uvedeného popisu webového logu vidíme, že ukládá informace o pohybu uživatelů na jednom serveru. Na základě těchto uložených informací můžeme určit pohyb uživatele na daném serveru a získanou informaci pak dále vyhodnotit. Bohužel i webový log má své nevýhody. Prvním problémem je, že není schopen uložit kompletní pohyb uživatele na serveru. Pokud se např. uživatel vrátí na stránku, které je uložená v cache paměti prohlížeče, není tato informace zaznamenána v logu. Důvodem je, že nedojde k dotazu na server, jelikož daná stránka je stále uložena v paměti prohlížeče, a proto nemůže server do webového logu nic zapsat. Dalším problémem je, že HTTP je bez stavový protokol a tato skutečnost vede k tomu, že není možné zaznamenat dobu, kterou uživatel stráví prohlížením dané stránky. Některé z uvedených problémů se dají odstranit, resp. zmírnit použitím dalších nástrojů. Pro sledování konkrétního uživatele můžeme např. využít cookies soubory. Cookies je informace uložené serverem ve webovém prohlížeči klienta. Problémem je, že cookies vyžadují spolupráci uživatele, a tedy na možnost jejich použití nemůžeme spoléhat.

Další možností jak získat potřebná data ze serveru, je odposlouchávání paketů (packet sniffing). Tato technika monitoruje příchozí a odchozí TCP/IP pakety. Je schopná poskytnout shodné informace s webovým logem, pokud je sniffer umístěn před webový server. Pomocí této techniky je také možné sbírat data z užití z více webů současně. To je možné vhodným umístěním snifferu v síti.

Data od klienta

Jsou zaměřena na sledování chování uživatele během procházení jedné nebo i více webových stránek. Existují dvě metody pro sběr dat od klienta. První možností je použití „remote agents“, např. Javascript, nebo Java applets. Druhou možností je použití upraveného webového prohlížeče. Prohlížeč je modifikován takovým způsobem, aby měl lepší podporu pro sběr informací. Nevýhodou obou těchto metod je, že vyžadují spolupráci uživatele.

Při použití první metody sledujeme chování uživatele na jedné stránce. Oproti sběru dat ze serveru odstraňuje tato metoda problém s cachováním a také výrazně zjednodušuje problém identifikace sezení uživatele. I tato metoda však má své nedostatky. Při použití Java apletů např. stále nemáme informaci o době, jakou uživatel stráví prohlížením stránky. Javascript pak např. není schopen zaznamenat obnovení stránky, nebo použití tlačítka „zpět“.

Druhá metoda umožňuje sledovat chování uživatele na více webových stránkách. Hlavní překážkou však je, jak přesvědčit uživatele, aby použil upravený prohlížeč. Nejčastěji je proto uživateli nabídnuta určitá odměna za používání upraveného prohlížeče. Upravený prohlížeč je schopen zachytit jednak použití tlačítek zpět, nebo obnovení stránky, ale také informaci o době prohlížení stránky. Metoda je tedy velice výhodná pro sběr informací, avšak musíme přesvědčit uživatele, aby daný prohlížeč používal.

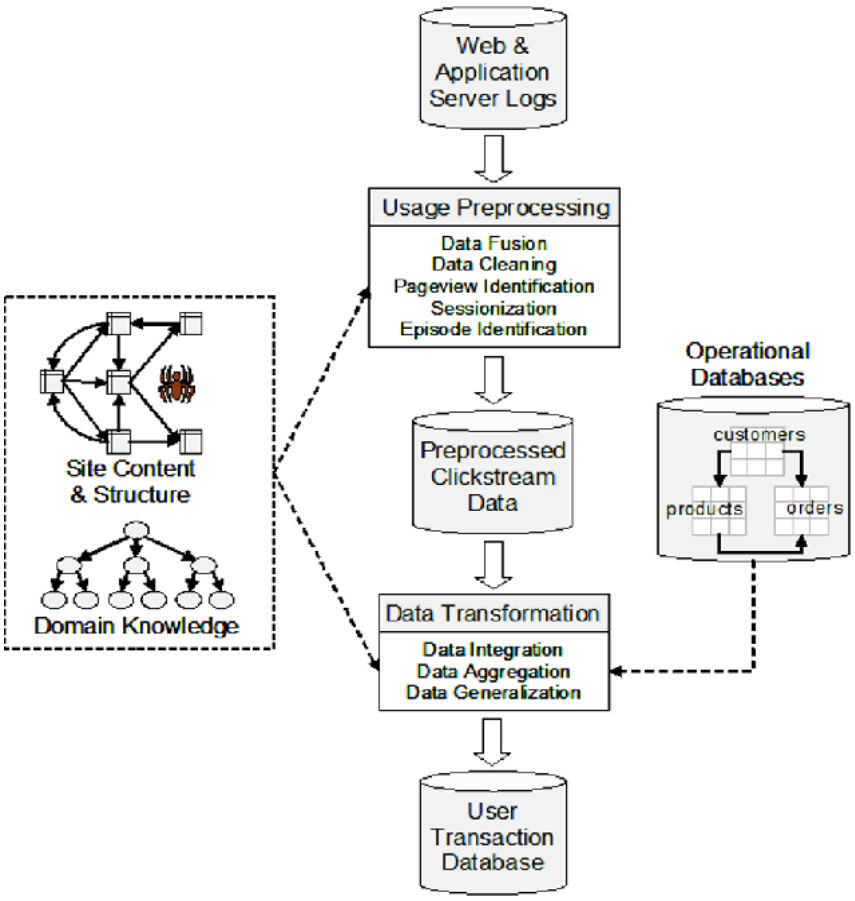
Data z Proxy

Jelikož webová proxy zastává funkci prostředníka mezi prohlížečem klienta a webovým serverem, může být proxy použita ke sběru dat. Je schopná sbírat data o prohlížení webových serverů skupinou uživatelů. Získaná data pak mohou posloužit pro získání znalostí o chování této skupiny uživatelů. Jedná se o skupinu uživatelů, kteří používají stejný proxy server. Kromě sběru dat ze serveru a od klienta je toto další možnost jak získat data potřebná pro dolování znalostí z užití Webu.

3.2 Předzpracování dat

Předzpracování dat je velmi důležitý úkol v každé aplikaci založené na dolování dat. Jejím úkolem je vytvořit ze vstupních dat takovou množinu dat, která bude vhodná pro algoritmy dolování dat. Transformace vstupních dat je často náročná na spotřebu času i výpočetního výkonu. Transformace dat je však nutná pro získání užitečných vzorů. Proces transformace dat může zahrnovat předzpracování dat, integraci dat z více zdrojů a upravení integrovaných dat do podoby vhodné pro další zpracování pomocí konkrétního algoritmu dolování dat. Tento proces se souhrnně nazývá příprava dat (data preparation).

Jak bylo uvedeno, data je nutné připravit pro algoritmy dolování dat. Jednou z těchto fází je



Obrázek 3-3 kroky přípravy dat pro Získávání znalosti z užití webu (Web usage mining) (převzato z [4])

předzpracování dat (viz Obrázek 3-3). Proces předzpracování dat se sám skládá z několika kroků. Některé hlavní kroky procesu předzpracování dat si přiblížíme v následujících podkapitolách.

3.2.1 Čištění dat

Proces čištění dat je obvykle specifický pro každý Web. Součástí samotného procesu jsou obvykle úkoly spojené s odstraněním nedůležitých záznamů v logovacím souboru, odstraněním některých datových polí v logu a smazáním záznamů, které byly vytvořeny roboty, jako je crawler, spider, atd.

Odstraněním nedůležitých záznamů v logovacím souboru je myšleno odstranění automaticky generovaných požadavků, tedy požadavků vytvořených např. v rámci načítání webové stránky, nikoliv však uživatelem jako reakce na kliknutí na odkaz. Takovéto záznamy jsou nepodstatné pro analýzu a úlohy dolování dat, a proto je nutné je odstranit. Typicky se jedná o požadavky na objekty, jako jsou obrázky CSS styly, zvukové soubory, atd. Volba objektů, které lze bezpečně odstranit, je však vždy závislá na úloze, kterou chceme s výslednými daty provádět. Např. obrázky jsou v mnoha případech považovány za nedůležité. Můžeme však chtít provádět analýzu prohlížení a stahování fotografií ve webové galerie a v tomto případě by odstranění záznamů obsahujících požadavky na obrázky bylo chybou. Fáze odstranění nedůležitých záznamů v procesu čištění dat slouží k redukci velikosti webového logu. Tento krok může snížit velikost webového logu až o polovinu, což je velmi důležité z hlediska toho, že Webový log může obsahovat obrovské množství záznamů.

Další částí čištění dat je identifikace a odebrání datových polí z webového logu, které nebudou v rámci prováděné úlohy poskytovat užitečné informace. V některých případech je tak možné odstranit pole udávající počet přenesených bytů, atd. Odebírání datových polí z webového logu je tedy závislé na prováděné úloze dolování dat, a také na formátu logu.

Posledním úkolem v procesu čištění dat, o kterém se zmíníme, je odstranění záznamů způsobených tzv. crawlery a spidery. Souhrnně je budeme nazývat roboty. Těmito roboty bývají programy, které pro webové vyhledávače prochází webové stránky. Soubor logu často obsahuje množství záznamů způsobených těmito roboty. Tito roboti se svým chováním liší od lidských uživatelů, a proto je nutné odstranit záznamy vytvořené činností robotů, aby nedošlo ke zkreslení výsledků chování uživatelů získaných analýzou. K identifikaci robotů se dá využít pole User Agent z webového logu. Z hlediska identifikace a odstranění robotů je výhodné udržovat seznam známých crawlerů. Roboti, které se nepodaří odstranit na základě tohoto seznamu, mohou být odstraněni po provedení úkolu identifikace sezení. A to z toho důvodu, že někteří roboti začínají své sezení tím, že se snaží získat přístup k souboru „robots.txt“ v kořenovém adresáři serveru. Nicméně zdaleka ne všichni roboti se řídí tímto chováním. Někteří se záměrně maskují jako běžní uživatelé, a proto může být nutné pro jejich identifikaci použít heuristické metody.

3.2.2 Identifikace uživatele

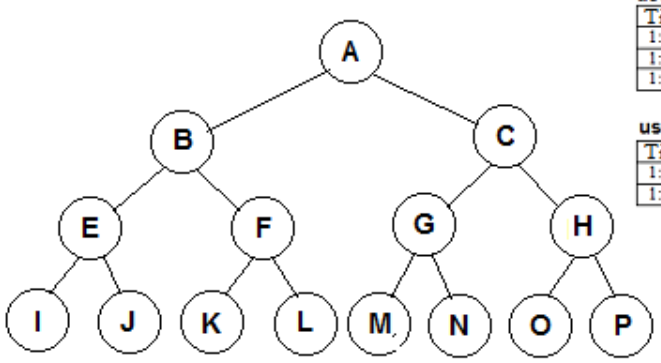
Analýza užití Webu nevyžaduje znalost identity uživatele, nicméně je nezbytné rozlišit konkrétní uživatele mezi ostatními uživateli. Jelikož uživatel může navštívit daný Web vícekrát, server ukládá do webového logu několik sezení pro každého uživatele. V ideálním případě by bylo možné každého uživatele rozlišit na základě poskytnutých registračních údajů. Problémem je, že na většinu webových stránek uživatel vstupuje anonymně, a proto v důsledku chybějící autentifikace návštěvníků se hledaly jiné možnosti jak návštěvníky jednoznačně rozlišit.

Jednou z možností je využití klientských cookies, což jsou krátké textové zprávy uložené na straně klienta. Cookie si většinou ukládá u klienta sám server a používá uložené informace k identifikaci uživatele. Nicméně mnoho uživatelů se obává zneužití informací v cookie. A proto kvůli obavám z narušení soukromí blokují použití cookies.

Pro identifikaci uživatele nepostačuje ani IP adresa. Důvodem je velká rozšířenost používání proxy serverů. Několik uživatelů tak může přistupovat na Web se stejnou IP adresou. V důsledku toho, že více uživatelů může přistupovat na Web se stejnou IP adresou, nejsme schopni rozlišit, zda dva přístupy na danou webovou stránku jsou ve skutečnosti generovány jedním, nebo dvěma uživateli. Nicméně i bez autentifikace nebo využití cookie je stále možné s poměrně velkou přesností identifikovat uživatele, a to díky kombinaci IP adresy a ostatních dostupných informací, kterými jsou např. pole user agent a referrer.

Pro identifikaci uživatele se tak využívá kombinace IP adresy a pole user agent. Jak znázorňuje obrázek 3-4, jde za použití této kombinace polí rozlišit uživatele se stejnou IP adresou. Pokud mají dva záznamy ve webovém logu stejnou IP adresu, ale liší se v poli user agent, jde o dva rozdílné uživatele. Ve výjimečném případě se může stát, že jeden uživatel vstoupí na stejnou webovou stránku se dvěma různými prohlížeči na stejném operačním systému, takovéto chování je však málo časté. Proto i když popsaná metoda s danou možností nepočítá, dává poměrně přesné výsledky při identifikaci uživatelů. Pro ještě přesnější rozlišení uživatelů je možné využít pole referrer. Obrázek 3-4 ukazuje, že jsme ze záznamů v logu určili 4 uživatele. Pomocí pole referrer, můžeme sestavit posloupnost jejich kroků. Díky tomu a znalosti o topologii Webu můžeme zjistit, zda uživatel mohl dané kroky provézt. Tedy pokud je pole referrer prázdné a neexistuje cesta z poslední stránky na aktuální stránku, můžeme dojít k závěru, že se pravděpodobně jedná o dva rozdílné uživatele, viz obrázek 3-4.

Time	IP	URL	Referrer	User agent
0:01	91.195.107.31	A	-	MSIE 9.0; WinXP
0:09	91.195.107.31	B	A	MSIE 9.0; WinXP
0:10	74.125.18.149	A	-	Mozilla/5.0; WinXP
0:12	74.125.18.149	C	A	Mozilla/5.0; WinXP
0:22	91.195.107.31	E	B	MSIE 9.0; WinXP
0:25	74.125.18.149	G	C	Mozilla/5.0; WinXP
0:33	91.195.107.31	J	E	MSIE 9.0; WinXP
0:58	74.125.18.149	M	G	Mozilla/5.0; WinXP
1:10	91.195.107.31	C	-	MSIE 9.0; WinXP
1:16	91.195.107.31	H	C	MSIE 9.0; WinXP
1:20	91.195.107.31	O	H	MSIE 9.0; WinXP
1:26	91.195.107.31	A	-	Mozilla/5.0; WinXP
1:30	91.195.107.31	C	A	Mozilla/5.0; WinXP



user 1

Time	IP	URL	Referrer	User agent
0:10	74.125.18.149	A	-	Mozilla/5.0; WinXP
0:12	74.125.18.149	C	A	Mozilla/5.0; WinXP
0:25	74.125.18.149	G	C	Mozilla/5.0; WinXP
0:58	74.125.18.149	M	G	Mozilla/5.0; WinXP

user 2

Time	IP	URL	Referrer	User agent
0:01	91.195.107.31	A	-	MSIE 9.0; WinXP
0:09	91.195.107.31	B	A	MSIE 9.0; WinXP
0:22	91.195.107.31	E	B	MSIE 9.0; WinXP
0:33	91.195.107.31	J	E	MSIE 9.0; WinXP

user 3

Time	IP	URL	Referrer	User agent
1:10	91.195.107.31	C	-	MSIE 9.0; WinXP
1:16	91.195.107.31	H	C	MSIE 9.0; WinXP
1:20	91.195.107.31	O	H	MSIE 9.0; WinXP

user 4

Time	IP	URL	Referrer	User agent
1:26	91.195.107.31	A	-	Mozilla/5.0; WinXP
1:30	91.195.107.31	C	A	Mozilla/5.0; WinXP

Obrázek 3-4 Příklad Identifikace uživatele

3.2.3 Identifikace sezení

Identifikace sezení je proces provádějící rozdělení záznamů o aktivitě každého uživatele do sezení. Každé sezení představuje jednu návštěvu zkoumaného Webu. Jelikož je však většina Webů přístupná bez nutnosti autentifikace, je nutné nejprve provézt identifikaci uživatele. Pro samotnou identifikaci sezení je pak nutné použít heuristické metody. Důvodem je, že na Webu většinou nebývají žádné zabudované mechanismy pro identifikaci sezení.

Heuristické metody pro identifikaci sezení se obecné dělí na dvě kategorie, časově orientované a strukturně orientované. Časově orientované heuristické metody jsou založené na využití časového prahu pro rozdělení záznamů do sezení. Naproti tomu strukturně založené heuristiky využívají pro rozlišení sezení např. informaci uloženou v poli referrer webového logu. Vytvořeno však bylo mnoho variant jak časově, tak i strukturně orientovaných heuristických metod. Vliv různých heuristik na úkoly spojené se získáváním znalosti z užití Webu byl analyzován v [4]. Pro demonstraci toho, jak mohou metody pro identifikaci sezení fungovat si dvě varianty časově orientovaných heuristik a jednu strukturně orientovanou ukážeme.

U první časově orientované varianty nesmí doba sezení přesáhnout zvolený práh. Toho je dosaženo tak, že vždy vezmeme časové razítko prvního záznamu v sezení S a časové razítko záznamu, pro nějž chceme určit, zda do daného sezení náleží. Provedeme rozdíl časového razítka zkoumaného záznamu a prvního záznamu, a pokud je výsledek menší než zvolený práh, pak daný záznam přiřadíme do sezení S. Jinak je tento záznam prvním záznamem nového sezení.

Druhá časově orientovaná varianta je založená na myšlence, že doba prohlížení stránky nesmí přesáhnout zvolený práh. Tedy podobně jako v předchozí variantě vezmeme dvě časová razítka a provedeme jejich rozdíl. Oproti předchozí variantě však budeme brát časová razítka dvou po sobě jdoucích záznamů, tedy poslední záznam v sezení a záznam, který chceme zařadit do sezení. Pokud je získaná hodnota větší než zvolený práh, vložíme záznam do nového sezení. Jinak je záznam vložen do aktuálního sezení. Časová hodnota prahu se v mnoha aplikacích ustálila na hodnotě 30 minut.

Poslední uváděnou variantou a první variantou orientovanou strukturně je metoda založená na využití pole referrer. Tato metoda říká, že požadavek A je zařazen do sezení S, pokud hodnota v poli referrer požadavku A je obsažena v sezení S. V opačném případě je požadavek A použit jako první záznam v novém sezení. U této metody může nastat situace, kdy požadavek A může podle uvedeného postupu náležet do více sezení. A to z důvodu existence více sezení, ve kterých bylo přistupováno ke stránce z pole referrer požadavku A. Řešením této vzniklé situace může být např. přiřazení požadavku A do nejnovějšího sezení, které splňuje podmínku této metody.

Time	IP	URL	Referrer
0:01	91.195.107.31	A	-
0:09	91.195.107.31	B	A
0:19	91.195.107.31	C	A
0:25	91.195.107.31	E	C
0:31	91.195.107.31	F	E
0:35	91.195.107.31	G	F
1:16	91.195.107.31	A	-
1:19	91.195.107.31	B	A
1:26	91.195.107.31	C	A
1:30	91.195.107.31	J	G
1:36	91.195.107.31	D	B
1:50	91.195.107.31	P	D

sezení 1 (session 1)

Time	IP	URL	Referrer
0:01	91.195.107.31	A	-
0:09	91.195.107.31	B	A
0:19	91.195.107.31	C	A
0:25	91.195.107.31	E	C

sezení 2 (session 2)

Time	IP	URL	Referrer
0:31	91.195.107.31	F	E
0:35	91.195.107.31	G	F

sezení 3 (session 3)

Time	IP	URL	Referrer
1:16	91.195.107.31	A	-
1:19	91.195.107.31	B	A
1:26	91.195.107.31	C	A
1:30	91.195.107.31	J	G
1:36	91.195.107.31	D	B

sezení 4 (session 4)

Time	IP	URL	Referrer
1:50	91.195.107.31	P	D

Obrázek 3-5 Příklad identifikace sezení s časově orientovanou heuristikou

Time	IP	URL	Referrer
0:01	91.195.107.31	A	-
0:09	91.195.107.31	B	A
0:19	91.195.107.31	C	A
0:25	91.195.107.31	E	C
0:31	91.195.107.31	F	E
0:35	91.195.107.31	G	F
1:16	91.195.107.31	A	-
1:19	91.195.107.31	B	A
1:26	91.195.107.31	C	A
1:30	91.195.107.31	J	G
1:36	91.195.107.31	D	B
1:50	91.195.107.31	P	D

sezení 1 (session 1)

Time	IP	URL	Referrer
0:01	91.195.107.31	A	-
0:09	91.195.107.31	B	A
0:19	91.195.107.31	C	A
0:25	91.195.107.31	E	C
0:31	91.195.107.31	F	E
0:35	91.195.107.31	G	F

sezení 2 (session 2)

Time	IP	URL	Referrer
1:16	91.195.107.31	A	-
1:19	91.195.107.31	B	A
1:26	91.195.107.31	C	A
1:30	91.195.107.31	J	G
1:36	91.195.107.31	D	B
1:50	91.195.107.31	P	D

Obrázek 3-6 Příklad identifikace sezení s časově orientovanou heuristikou

Obrázek 3-5 a obrázek 3-6 ukazují příklad identifikace sezení pomocí popsaných časově orientovaných heuristik. V obou případech je hodnota prahu nastavena na 30 minut. Jak můžeme vidět na Obrázek 3-5 aplikací první popsané varianty vznikly 4 sezení. Naproti tomu aplikací druhé časově orientované heuristiky, která byla popsána v textu výše, jsme získali pouze dvě sezení, viz obrázek 3-6. Jak můžeme vidět, v prvním případě sezení trvají 30 minut, tedy délku prahu. V druhém případě jsou sezení rozdělena, pokud doba mezi výskyty záznamů přesáhne dobu definovanou prahem.

Posledním příkladem je obrázek 3-7, který zobrazuje identifikaci sezení pomocí strukturně orientované heuristiky, která byla taktéž popsána v textu výše. Vidíme, že podobně, jak znázorňuje obrázek 3-6, jsme identifikovali dvě sezení. Nicméně při bližším pohledu vidíme, že dané sezení nejsou úplně shodná. Oproti časově orientované heuristice je zde záznam v čase 1:30 přiřazen do sezení 1, protože pole referrer tohoto záznamu je obsaženo v sezení 1 a nikoliv v sezení 2.

Time	IP	URL	Referrer
0:01	91.195.107.31	A	-
0:09	91.195.107.31	B	A
0:19	91.195.107.31	C	A
0:25	91.195.107.31	E	C
0:31	91.195.107.31	F	E
0:35	91.195.107.31	G	F
1:16	91.195.107.31	A	-
1:19	91.195.107.31	B	A
1:26	91.195.107.31	C	A
1:30	91.195.107.31	J	G
1:36	91.195.107.31	D	B
1:50	91.195.107.31	P	D

sezení 1 (session 1)

Time	IP	URL	Referrer
0:01	91.195.107.31	A	-
0:09	91.195.107.31	B	A
0:19	91.195.107.31	C	A
0:25	91.195.107.31	E	C
0:31	91.195.107.31	F	E
0:35	91.195.107.31	G	F
1:30	91.195.107.31	J	G

sezení 2 (session 2)

Time	IP	URL	Referrer
1:16	91.195.107.31	A	-
1:19	91.195.107.31	B	A
1:26	91.195.107.31	C	A
1:36	91.195.107.31	D	B
1:50	91.195.107.31	P	D

Obrázek 3-7 Příklad identifikace sezení se strukturně orientovanou heuristikou

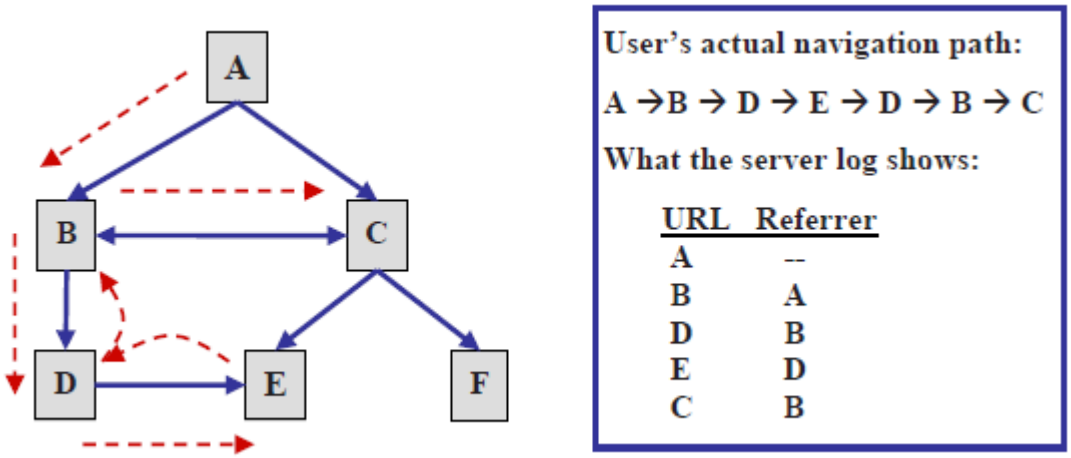
Kromě identifikace sezení je možné provézt tzv. identifikaci epizody. Identifikace epizody se provádí v závěrečné fázi předzpracování a jejím úkolem je zaměřit se v rámci každého sezení jen na relevantní podmnožinu prohlížených stránek. Epizoda tak reprezentuje podmnožinu prohlížených stránek, které jsou určitým způsobem, ať už sémanticky, nebo funkcionálně, podobné. Abychom mohli provézt identifikaci epizody, je tedy nutné provézt klasifikaci prohlížených stránek do kategorií. Proces klasifikace stránek do kategorií může být automatický nebo semi-automatický.

3.2.4 Kompletace cesty

Kompletace cesty uživatele při návštěvě Webu je dalším důležitým úkolem předzpracování dat. Tato úloha se provádí převážně po dokončení identifikace sezení. Kompletace cesty uživatele se poté provádí v rámci každého sezení neboli jedné návštěvě analyzovaného Webu. Důvodem, proč se musí kompletace cesty provádět, je to, že ne všechny údaje o prohlížení stránek jsou uloženy ve webovém logu. Stránky, na které již uživatel v rámci sezení vstoupil, jsou uloženy na straně klienta nebo proxy v cache paměti a opětovným přístupem na takovéto stránky tak již nevznikne požadavek na server. To

způsobí, že tento druhý vstup na stránku nebude zaznamenán ve webovém logu. Takovéto chybějící záznamy je možné na základě použití heuristických metod odvodit. Heuristické metody se v tomto případě spoléhají na znalost struktury webu a na pole referrer v logu.

Jako příklad poslouží obrázek 3-8, který zobrazuje strukturu analyzovaného webu, záznamy z webového logu a cestu uživatele webem. Můžeme vidět (červené tečkované čáry), že poté co si uživatel prohlédl stránku E, tak se pomocí tlačítka zpět webového prohlížeče vrátil na stránku D a poté na stránku B. Navrácení se na tyto dvě stránky se nezobrazí ve webovém logu, protože stránky byly uloženy v cache paměti. Log dále ukazuje, že další požadavek byl na stránku C a v poli referrer je B. V logu tedy není zaznamenána informace o navigaci ze stránky E na B. Vzhledem ke znalosti struktury webu lze odvodit, že chybějící záznamy představují navigaci z E->D a z D->B. Důležité je poznamenat, že zvolená varianta dokončení cesty uživatele není jedinou možnou cestou. Volba variant pro dokončení cesty závisí na zvolené heuristice.[5]



Obrázek 3-8 Kompletace cesty (převzato z [5])

3.2.5 Integrace dat

Výsledkem všech úkolů fáze předzpracování dat je množina uživatelských sezení. Každé sezení představuje posloupnost uživatelem prohlížených stránek. Nicméně protože chceme poskytnout fázi nalezení vzorů co možná nejlepší a nejdůležitější informace, které by mohly pomoci k nalezení vzorů, je nutné provézt integraci dat z více zdrojů. Např. při analýze webu internetového obchodu spojíme data získaná z fáze předzpracování s uživatelskými daty, zahrnujícími např. historii nákupů uživatele, a s informacemi o poskytovaných produktech a kategoriích získaných z operační databáze. Spojení těchto informací může vézt k odhalení důležitých obchodních a marketingových informací. Obecně jsou tato spojená data uložena v konečné transakční databázi a uložena v datovém skladu.

Až do teď jsme si v této kapitole představily metody pro předzpracování dat, které jsou specializované na zpracování dat z webových logů. Nicméně kromě těchto specializovaných metod pro předzpracování musíme na data aplikovat i běžné metody pro předzpracování dat.

Prvním úkolem je odstranění chybějících hodnot. Problém chybějících hodnot se vyskytuje neustále a pouze ve výjimečných případech může být tato skutečnost přínosná. Typicky se jedná o chybějící hodnoty v databázi, ale obecně můžou hodnoty chybět v jakémkoliv zdroji dat. Protože chybějící hodnoty nejsou přínosné, existuje několik přístupů jak se s těmito hodnotami vypořádat.

- **Odstranění n-tice** – První možností je odstranění celého řádku, kterému v některém sloupci chybí hodnoty. Tento přístup se však ukázal jako nepříliš prospěšný, protože odstranění celého řádku může vézt k tomu, že výsledná množina dat bude zkreslená. Odstranění celého řádku je výhodné pouze tehdy, pokud řádek obsahuje více sloupců s chybějící hodnotou.
- **Manuální nahrazení** – Tento přístup by mohl dávat uspokojivé výsledky, nicméně problémem je jeho příliš velká časová náročnost. Principem této metody je předpoklad, že uživatel má určité znalosti, které by mu mohly pomoci při nahrazování chybějící hodnoty.
- **Globální konstantou** – Jedná se o automatickou náhradu chybějící hodnoty. V databázích by se typicky jednalo o hodnotu NULL, pro numerický atribut by to mohla být např. hodnota 0. U takto doplněných hodnot je možnost, že pokud je hodnota mimo rozsah platných hodnot daného atributu, může být v dalších úkolech předzpracování dat ignorována, protože bude chápána jako odlehlá hodnota. Problémem však zůstává, že pokud by takovýchto chybějících hodnota bylo více, tak tato uměle zavedená hodnota by se mohla stát významnou a ovlivnila by negativně výslednou množinu dat.
- **Průměrnou hodnotou atributu** - V tomto případě bude chybějící hodnota nahrazena průměrnou hodnotou atributu, která bude spočítána ze všech ostatních hodnot v daném sloupci.
- **Průměrem hodnot n-tic patřících do téže třídy** – Jde o podobný přístup jako o předchozí varianty. Pouze průměr se počítá jen z hodnot řádků patřících do stejné kategorie.
- **Nejpravděpodobnější hodnotou** – U této varianty se pro určení chybějící hodnoty využije hodnot ostatních atributů v n-tici. Jde tedy o řešení úloh, jako jsou predikce, případně klasifikace. Chybějící atribut je v takovémto případě hledaným atributem.

Kromě odstranění n-tice a manuálního nahrazení se všechny ostatní varianty řadí do automatické náhrady. Automatická náhrady chybějící hodnoty však vždy představuje určité ovlivnění množiny dat. Jako nejlepší varianta se tak jeví poslední možnost, která provádí náhradu s pomocí klasifikace nebo predikce a využívá v porovnání s ostatními variantami více informací o modifikovaném záznamu.

Dalším úkolem je identifikace odlehlých hodnot. Odlehlé hodnoty jsou hodnoty, které leží na okraji datového rozsahu nebo se určitým způsobem vymykají trendu ostatních dat. Identifikace těchto

hodnot je důležitá, neboť tyto hodnoty mohou představovat chybu v datech. V některých případech může být tato hodnota i platná, nicméně pro některé statistické metody je přítomnost těchto hodnot problémová a tyto metody poté mohou poskytovat nepřesné výsledky. K identifikaci odlehlých hodnot mohou být použity jak grafické metody, tak i metody numerické. Příkladem grafických metod může být histogram, na kterém můžeme snadno odhalit odlehlé hodnoty.

Posledním úkolem, o kterém se zmíníme, je normalizace a standardizace dat. Proměnné v datech se často velmi liší od ostatních. To vede na velké rozsahy hodnot v těchto datech. S velkými rozsahy hodnot mohou mít některé algoritmy pro dolování dat problémy, což by mohlo způsobit zkreslené výsledky. Proto by se měla provádět normalizace numerických proměnných, aby se sjednotil rozsah vlivu proměnných na výsledek. Existuje množství technik pro normalizaci. V tomto textu si uvedeme Min-Max normalizaci.

Min-Max normalizace pracuje na principu nalezení rozdílu mezi hodnotou, která má být normalizována, a minimální hodnotou v poli. A následně upravení tohoto rozdílu pomocí rozsahu hodnot na hodnotu reprezentující normalizovanou hodnotu původní proměnné. Vzorec pro Min-Max normalizaci vypadá následovně.

$$X^* = \frac{X - min(X)}{range(X)} = \frac{X - min(X)}{max(X) - min(X)}$$

Rovnice 3-1 Vzorec pro Min-Max normalizaci

Min-Max normalizované hodnoty budou mít tedy rozsah od hodnoty 0 do hodnoty 1. Jediná výjimka nastane, pokud se objeví nové hodnoty, které budou ležet mimo rozsah (range), pomocí kterého byli normalizované hodnoty vypočítány.

3.3 Nalezení vzorů

Cílem fáze nalezení vzorů je pochopit, jak se uživatelé chovají na zkoumaném Webu. Existuje velké množství metod, které mohou být použity pro nalezení vzorů z dat užití webu. Tyto metody mohou být založeny na statistice, strojovém učení, rozpoznávání vzorů, atd. Tyto metody se liší i co do složitosti. Od relativně jednoduchých až po výpočetně velice náročné metody. Pro získání přehledu o tom, jak je Web používán, je na data většinou aplikována více než jedna metoda. Některé metody pro dolování dat si rozebereme v následujícím textu.

3.3.1 Statistická analýza

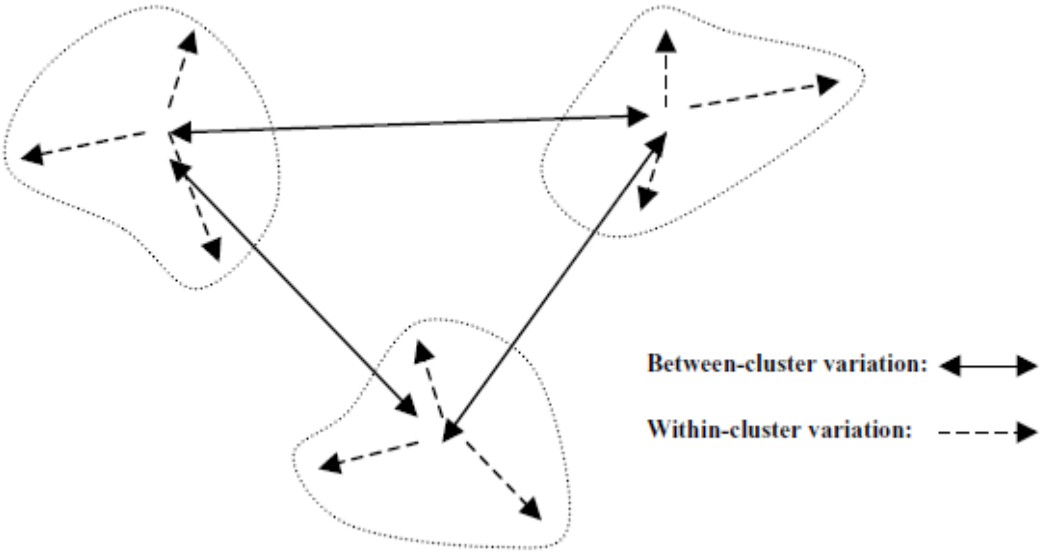
Statistické metody se řadí mezi nejčastěji používané metody pro získávání znalostí o návštěvnících Webu. Statistická analýza se zároveň řadí mezi relativně jednoduché a výpočetně méně náročné metody získávání znalostí. Využitím statistické analýzy můžeme získat mnoho statistických pohledů, např. medián, frekvence, na proměnné jako jsou počet zhlédnutí stránky, čas strávený prohlížením

stránky, atd. Přestože statistická analýza poskytuje informace získané pomocí analýzy, které nezkoumá data příliš do hloubky, i tyto informace mohou být potenciálně užitečné pro vylepšování struktury Webu, bezpečnosti, pro marketingové účely, atd. Statistickou analýzu využívá mnoho nástrojů pro analýzu webového provozu.

3.3.2 Shlukování

Shlukování, neboli clustering, se zabývá rozdělováním záznamů do tzv. clusterů. Cluster je kolekce záznamů, které jsou si podobné a nejsou podobné záznamům v jiných clusterech. Jedná se tedy o techniku dolování dat založenou na rozdělení množiny záznamů do skupin, v níž mají všechny záznamy podobné charakteristiky. Obrázek 3-9 zobrazuje požadovaný cíl všech metod shlukování, tedy rozdělení do clusterů, kde podobnost záznamů v rámci clusteru (within-cluster variation) je velká a podobnost záznamů mezi clustery (between-cluster variation) je nízká. Technika shlukování je často používána jako předběžný krok v procesu dolování dat. Získané clustery jsou poté použity jako vstupy pro další techniku. V oblasti získávání znalostí z užití Webu existují dva zajímavé clustery. Jedná se o user cluster a page cluster.

User cluster neboli shlukování uživatelských záznamů, jako jsou sezení a transakce, je jednou z nejčastěji používaných analyzačních úloh v rámci získávání znalostí z užití Webu. Shlukování uživatelů provádí vytváření skupin uživatelů na základě toho, že si tito uživatelé prohlížejí či prohledávají stejné stránky. Takovéto znalosti jsou užitečné zejména při odvozování demografických informací o uživateli, na jejichž základě je pak uživateli předkládána reklama, nebo poskytován personalizovaný obsah Webu pro uživatele se stejnými zájmy. Bližší analýza skupin uživatelů založených na jejich demografických vlastnostech, jako jsou věk, pohlaví, příjem, atd., může poskytnout cenné obchodní informace. Tato forma shlukování byla také využita k vytvoření webových uživatelských komunit pro uživatele s podobnými zájmy, více viz [6].



Obrázek 3-9 Ukázka techniky Shlukování

Page cluster neboli shlukování stránek může být provedeno na základě obsahu webových stránek nebo dat o užití stránek. Při použití shlukování na základě obsahu může být výsledkem kolekce stránek, nebo produktů vztahujících se ke stejnému tématu, nebo kategorii v případě produktů. Při využití informací o užití pro shlukování je možné položky, které jsou často prohlíženy nebo kupovány, společně automaticky organizovat do skupin. Shlukování stránek na základě obsahu je např. velmi výhodné pro vyhledávače, které na základě shlukování mohou vytvořit stránka s odkazy na podobné zdroje.

3.3.3 Asociační pravidla

Dolování asociačních pravidel (association rules) je základním úkolem v oblasti dolování dat. Tato technika byla poprvé představena v roce 1993 v [7]. Jejím cílem je najít všechny souběžné výskyty položek v datech. Tyto výskyty se nazývají asociace. V oblasti získávání znalostí z užití Webu mohou být položkami stránky nebo zboží. Klasickou aplikací dolování asociačních pravidel je analýza nákupního košíku, které umožňuje odhalit, jaké vztahy jsou mezi zbožím, které zákazníci kupují, např. tedy umožňuje zjistit, jaké položky jsou nejčastěji kupovány společně. Asociační pravidlo může vypadat např. takto **Sýr→ Pivo [support = 10%, confidence = 80%]**.

Toto pravidlo říká, že 10 % zákazníků si koupí sýr a pivo společně, a zákazníci, kteří si koupí sýr, si v 80 % případech koupí také pivo. Support a confidence jsou dvě měřítka udávající sílu pravidla nebo také jinak řečeno pravděpodobnost výskytu pravidla. Parametr support pravidla X→Y udává, v kolika procentech případů z množiny dat se vyskytuje X a zároveň Y. Parametr support tedy vyjadřuje, jak často, je pravidlo použitelné v množině dat T. Hodnota parametru se vypočítá následovně:

$$support = \frac{(X \times Y) \times počet}{n}$$

Rovnice 3-2 Rovnice pro výpočet parametru support

, kde n udává počet záznamů v množině T. Druhým parametrem je confidence, který udává, v kolika procentech případů, jestliže transakce obsahuje X, tak také obsahuje Y. Tento parametr je možné určit pomocí vzorce:

$$confidence = \frac{(X \times Y) \times počet}{X \times počet}$$

Rovnice 3-3 Rovnice pro výpočet parametru confidence

.

Confidence určuje předvídatelnost pravidla. Pokud má tento parametr příliš nízkou hodnotu, nemůžeme odvozovat nebo předvídat Y z X. Použitelnost pravidla s nízkou předvídatelností je omezená.

Tato technika může být použita k množství úkolů. Již bylo zmíněno, že nejčastějším využitím je analýza nákupního košíku, další možnou aplikací je využití pro optimalizaci navigace v rámci Webu. Např. pokud uživatelé často prohlížejí stránku A, a poté stránku B, můžeme přidat přímí odkaz ze stránky A na stránku B, což uživateli usnadní nalezení požadované informace.

Nejčastějším přístupem k nalezení asociačních pravidel je použití algoritmu Apriori. Tento algoritmus nachází skupiny položek, které se objevují často společně v jedné transakci. Tyto skupiny jsou nazývány frequent itemsets. Ze skupin, které splňují minimální požadovanou hodnotu parametru confidence, jsou následně generována asociační pravidla. Algoritmus Apriori vytváří frequent itemsets skupiny z transakcí. Tyto transakce jsou vytvářeny z výstupu fáze předzpracování dat a cílem je pro každého uživatele vytvořit transakce, které budou obsahovat shluky smysluplných odkazů. Existují 3 metody jak identifikovat takovéto transakce:

Time window – První metodou je časové okno, kdy v rámci transakce uchováváme odkazy, které se vyskytly v rámci zadaného časového okna. Tedy rozdíl času výskytu prvního odkazu v transakci a posledního odkazu v transakci není větší než zadaná délka časového okna. Kvůli nutnosti zadat délku časového okna se tato metoda řadí do metod řízených uživatelem. Každá transakce je zde definována jako trojice obsahující IP adresu klienta, identifikaci klienta a posledním prvkem je množina obsahující dvojice URL odkazu a čas výskytu odkazu.

Reference length – Tato metoda je založena na podobném principu jako Time window. Její rozdíl spočívá v tom, že každému odkazu přiřazuje kromě času výskytu také hodnotu udávající dobu strávenou na dané stránce. Tato hodnota slouží k rozlišení, zda se jedná o stránku s obsahem, nebo pouze o navigační stránku. Znovu je tedy nutné zadat hodnotu, která bude sloužit k rozlišení navigační a obsahové stránky, a tedy jedná se o metodu řízenou.

Maximal forward reference – Jedná se o metodu, která nevyžaduje řízení uživatelem. Tato metoda rozlišuje dva typy odkazů, backward reference (znovu navštívení zdroje) a forward reference (navštívení nového zdroje). Metoda sestavuje cesty uživatele (tzv. maximal forward reference) a tato cesta je ukončena v okamžiku výskytu backward odkazu. Množina transakcí je tak dána maximálními cestami (maximal forward) uživatele. Např. máme následující cestu uživatele {A, B, C, D, C, B, E, G, H, G, W, A, O, U, O, V}. Aplikací metody Maximal forward reference získáme následující množinu transakcí: {ABCD, ABEGH, ABEGW, AOU, AOV}.

Po získání transakcí můžeme pro získání asociačních pravidel aplikovat např. algoritmus Apriori. Jak již bylo uvedeno, jeho aplikací získáme množinu frequent itemsets. Takto získaná množina však stále obsahuje pravidla, která nejsou pro analýzu důležitá, a proto je vhodné aplikovat další metody pro odstranění nedůležitých pravidel z této množiny. Můžeme např. odstranit pravidla

obsahující stránku index, protože je zjevné, že existuje přímý odkaz mezi stránkou index a stránkou na druhé straně pravidla. Další možnosti zmenšení množiny poskytují tzv. silná pravidla, tedy pravidla, jejichž hodnota confidence je blízká hodnotě 1. Tato metoda je založena na redukci množiny pravidel obsahující stejné silné pravidlo jediným pravidlem. Tato metoda je blíže popsána v [8]. Jiný přístup je uveden např. v [9]. Cílem je tedy snížit množinu pravidel tak, aby zůstali pouze relevantní pravidla, a aby množina pravidel nebyla příliš rozsáhlá a byla snadno analyzovatelná.

3.3.4 Klasifikace

Klasifikace (classification) je úloha zabývající se mapováním datových položek na několik, předem určených tříd. Klasifikace je tedy proces, který probíhá ve dvou fázích. První fáze je tzv. fáze učení. V této fázi jsou vybrány vzorky dat a množinu těchto dat nazveme trénovací množinou. U dat z této množiny musíme předem znát, do jaké třídy patří. Vzorky dat z trénovací množiny jsou následně jako vstup předloženy klasifikátoru, jehož úkolem je zjistit klasifikační pravidla. Na základě zjištěných klasifikačních pravidel se poté každý objekt může s jistou pravděpodobností přiřadit do konkrétní třídy.

Druhá fáze je fáze testování. Stejně jako v první fázi jsou vybrány vzorky dat, u kterých je předem známo, do které třídy patří, a množina těchto dat je nazvána testovací množina. Vzorky dat jsou za použití naučených klasifikačních pravidel zařazovány do tříd. Následné se na základě znalosti, do které třídy daný vzorek dat patří, určí, v kolika procentech případů se povedlo vzorek dat přiřadit do správné třídy. Podle dosažené hodnoty v procentech se určí, zda se klasifikační pravidla mohou použít pro data, u kterých není známo, do jaké třídy patří, nebo se vrátíme k první fázi.

Klasifikace je tedy technikou řadící se do oblasti supervised learning. V oblasti Webu je cílem vytvořit profily uživatelů příslušejících do určité skupiny. To vyžaduje extrakci a zvolení takových vlastností, které nejlépe popisují vlastnosti zvolených tříd. Klasifikaci je možné provádět na základě mnoha algoritmů, jako např. k-nearest neighbor classifiers, support vector machines, rozhodovací strom, Bayesovská klasifikace, atd.

Klasifikace hraje v analýze webových aplikací důležitou roli z hlediska schopnosti rozdělit uživatele do skupin na základě množství různých metrik. Klasifikace může být např. použita pro internetové obchody, kde bude provedeno zařazení uživatele do skupin na základě toho, kolik peněz utratil v našem obchodě za určité období. Na základě toho se pak můžeme rozhodnout, kterým uživatelům nabízet více nabídek, protože je u těchto uživatelů větší pravděpodobnost nákupu zboží. Speciálním případem klasifikačních a predikčních systémů na Webu je tzv. recommender systems. Tyto systémy jsou široce využívané na Webu pro doporučování produktů a služeb uživatelům. Tyto systémy poskytují dvě důležité funkce. První funkce má za úkol pomoci uživatelům získat přehled v obrovském množství informací. Tuto funkci plní poskytováním doporučení na základě znalostí

konkrétního uživatele a jeho zájmů, návyků, atd. Tato funkce tedy např. doporučí uživatele stránku, jejíž obsah by mohl uživatele zajímat. Druhá funkce má za úkol pomocí zvýšit tržby podnikatele.

Klasifikační metody mohou být založeny na mnoha typech algoritmů, všechny se však porovnávají na základě pěti následujících kritérií:

- **Přesnost předpovědi** – udává, v kolika procentech případů daný model správně klasifikuje nová data. Pod pojmem nová data chápeme data, která nebyla obsažena v trénovací množině.
- **Robustnost** – udává, schopnost vytvořit správný model v případě, kdy data obsahují šum a chybějící hodnoty.
- **Stabilita** – udává schopnost vytvořit správný model pro velké množství dat.
- **Rychlost** – udává nejen složitost první fáze, tedy složitost vytvoření klasifikačních pravidel, ale i výpočetní složitost používání těchto pravidel.
- **Interpretovatelnost** – udává složitost daného model pro pochopení.

3.4 Analýza vzorů

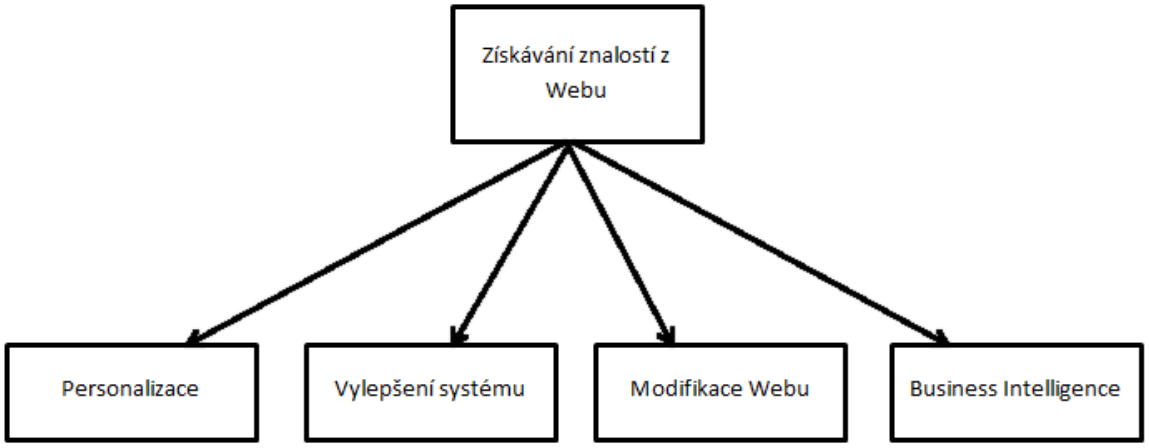
Analýza vzorů je posledním krokem v procesu získávání znalostí z užití Webu. Cílem tohoto kroku je transformovat získané vzory, pravidla a statistiky na znalosti. Transformace informace na znalost je složitá operace. Z tohoto důvodu poskytuje fáze analýzy vzorů nástroje pro usnadnění transformace informací na znalosti.

Nástrojem, který takovou transformaci zvládne, může být téměř jakýkoliv nástroj nebo filtr, který je možný použít na výstup algoritmů pro dolování dat. Jedním z nejčastějších nástrojů, který splňuje výše uvedené, jsou dotazovací mechanismy, jako např. SQL dotazy pro relační databázi, umožňující filtrování, seřazení a kombinaci různých dat. Lepší možností je využití nástroje OLAP (On-line Analytical Processing). OLAP poskytuje několik vrstev s různými stupni agregace dat v těchto vrstvách. Další často využívanou metodou je vizualizace výsledků pomocí grafů, tabulek, atd.

V oblasti získávání znalostí z užití Webu, ale obecně v každé oblasti dolování dat, je velkou výzvou rozhodnutí, které získané znalosti jsou potenciálně užitečné, a které znalosti jsou neužitečné. Rozhodnutí, které znalosti jsou užitečné a které nikoliv, však záleží především na konkrétní aplikaci. Např. aplikace pro marketingovou analýzu budou považovat za užitečné informace nejčastěji se vyskytující vzory. Naopak aplikace pro bezpečnostní analýzu budou za užitečné informace považovat nezvyklé vzory, které mají malou frekvenci výskytu. Metody pro identifikaci užitečných informací musí procházet velké množství vzorů. Důvodem je, že práh algoritmů pro nalezení vzorů je většinou nastavený nízko, aby se nalezly všechny potenciálně užitečné vzory. Pokud by byl tento práh nastavený na vyšší hodnotu, mohlo by dojít k situaci, kdy neobjevíme některé užitečné vzory.

3.5 Aplikace znalostí

Znalosti získané z informací o užití Webu je možné aplikovat v mnoha oblastech. Obrázek 3-10 zobrazuje nejčastější aplikace těchto znalostí. V této kapitole si přiblížíme způsob použití získaných znalostí v těchto oblastech.



Obrázek 3-10 Ukázka oblastí pro aplikaci znalostí získaných z užití Webu

Personalizace

Personalizace je jednou z nejběžnějších aplikací znalostí získaných z užití Webu. Jak její název napovídá, slouží pro přizpůsobení stránky uživateli. Mnoho webových stránek proto na základě získaných znalostí nabízí uživateli úpravu jeho profilu tak, aby mu lépe vyhovoval. Např. na základě informací o stránkách, které uživatel často prochází, můžeme uživateli nabídnout obsah, který by ho mohl zajímat. Další možností je cíleně zaměřit reklamy na uživatelově profilu podle jeho zájmů. Možností využití získaných znalostí v oblasti personalizace je mnoho.

Vylepšení systému

Výkon, bezpečnost a další atributy webových služeb jsou rozhodující pro spokojenost uživatele. Získávání znalostí z užití Webu poskytuje znalosti o provozu na zkoumaném Webu, které mohou být použity pro vytvoření politik pro Web caching, vyvažování zátěže, distribuci dat, atd. Bezpečnost je neustálým problémem ve webových aplikacích. I v této záležitosti však může dolování dat z užití Webu poskytnout vzory, které jsou užitečné při detekci podvodu, neoprávněného přístupu, atd.

Modifikace Webu

Atraktivita Webu, ať už z pohledu obsahu nebo struktury, je důležitá pro většinu aplikací. Získávání znalostí z užití Webu poskytuje vynikající odezvu uživatelského chování, což zároveň poskytuje návrhářům Webu informace, na jejichž základě lze provézt modifikace tak, aby Web lépe vyhovoval požadavkům uživatelů. Jasným příkladem je získání znalosti, že velká část uživatelů se přesouvá z jedné kategorie Webu do druhé a využívá při tom stránku, které je pouze navigační a neobsahuje žádný obsah. A proto přidáme přímý odkaz z první kategorie do druhé a usnadníme tak uživateli přesun mezi těmito kategoriemi.

Business Intelligence

Informace o tom, jak zákazníci používají zkoumaný Web je nezbytná informace pro obchodníky. Znalost chování uživatele je výhodná zejména v internetových obchodech, kde nám umožňuje na základě znalostí o zboží, které se často prodává současně, nabízet uživateli zboží, o které by mohl mít zájem. Potřebné znalosti se získávají převážně pomocí analýzy nákupního košíku. Znalosti se dají dále využít pro reklamní, marketingové účely, atd. Jedná se o velmi rozšířenou oblast aplikace znalostí z užití Webu.

3.6 Web log formáty

V této části se detailněji podíváme na nejčastější formáty logovacích souborů, které jsou využívány pro uchovávání záznamů o přístupu klientů na webové servery. Budou uvedeny čtyři nejčastější formáty těchto souborů, a to spolu s vysvětlením významu jednotlivých polí.

NCSA Common

Jde o základní formát, který je využíván jako základ pro mnoho dalších formátů. Uchovává základní HTTP přístupové informace. Tento formát bývá zkráceně označován jako CLF (Common Log Format) a jde o standardní textový formát, který se skládá ze sedmi polí. Tabulka 3-1 ilustruje formát CLF logu. Sloupec Host udává IP adresu, nebo doménové jméno klienta, který si vyžádal daný http zdroj. Rfc931 slouží pro uchování identity klienta, zjištěné pomocí protokolu ident. Jak však vidíme, v našem případě neobsahuje tento sloupec žádnou hodnotu, a proto je zde místo konkrétní hodnoty uvedena „-“ udávající, že hodnota chybí. Toto pole bývá v současnosti většinou prázdné. Dalším polem je Username, které je využíváno pro autentizaci klienta. Pole Date:time je časové razítko vzniku HTTP požadavku, uchovává den a čas vzniku požadavku. Pole Request se skládá ze tří

informací. První část udává HTTP metodu (nejčastěji GET, nebo POST), druhá část udává požadovaný zdroj (index.html a mease.jpg) a třetí část říká, jaká verze HTTP protokolu byla použita. Předposledním polem je Status code oznamující, zda byl požadavek úspěšně vyřízen, nebo zda došlo k chybě. Poslední pole udává počet přenesených bytů.

Host	Rfc931	Username	Date:time	Request	Status code	bytes
125.125.125.125	-	dsmith	[10/Oct/2006:21:15:05 +0500]	"GET /index.html HTTP/1.0"	200	1043
70.105.172.121	-	-	[20/Jun/2011:00:01:03 -0400]	"GET /mease.jpg HTTP/1.1"	200	2054

Tabulka 3-1 Ukázka NCSA Common logu

NCSA Combined

Je rozšířením předchozího NCSA Common formátu. Obsahuje tedy stejné informace, a navíc přidává tří dodatečné pole, které rozšiřují množinu uchovávaných informací a umožňují např. snazší identifikaci cesty klienta po daném Webu, jednodušší identifikaci spiderů, botů, atd. Rozšiřujícími poli jsou pole Referrer, User agent, Cookie. Pole Referrer udává URL adresu odkazující stránky, User agent identifikuje webový prohlížeč (resp. program), který požadavek zaslal. Cookie udává informace, které server zasílá zpět klientovi spolu s požadovaným zdrojem.

NCSA Separate

Jde o formát, který uchovává stejné hodnoty jako NCSA Combined formát, ale liší se způsobem uchování informace. Tento formát na rozdíl od předchozího neukládá informace v jediném souboru, nýbrž rozděluje informace do tří samostatných souborů. Tyto tři soubory jsou Common log, Referral log a Agent log. V Common log souboru jsou uchovávány informace, které jsou identické s informacemi, které uchovává CLF (NCSA Common) formát. Referral log obsahuje záznamy pro každou položku v Common log souboru. Obsahuje dvě pole, date:time a referrer pole. Význam těchto polí je stejný jako v předchozím textu. Posledním souborem je Agent log, který obdobně jako Referral log obsahuje záznam pro každou položku v Common logu a taktéž se skládá ze dvou polí. Těmito poli jsou date:time a user agent.

W3C Extended

Posledním uváděným formátem je W3C Extended Log Format. Protože je tento formát využíván Microsoft Internet Information Servery bývá označován jako IIS. Log soubor v tomto formátu se skládá z direktiv a záznamů. Direktivy jsou řádky začínající znakem „#“ a udávájí řídící informace. W3C Extended formát definuje direktivy :

Version – udává verzi extended formátu logovacícho souboru

Fields – je seznam specifikující jaké informace se budou v rámcí záznamů uchovávat

Software – identifikuje software, který vygeneroval daný log

Start-Date –udává datum a čas začátku logu

End-Date – udává datum a čas konce logu

Date – informace o datu, kdy byl následující záznam přidán

Remark – komentář k logu

Direktivy Version a Fields jsou povinné a musí být uvedeny na začátku každého logu. Záznamy jsou řádky řádky uchovávající informace o HTTP požadavcích. Jejich formát je proměnlivý a závísí na direktivě Fields. W3C Extended Log Format je tedy nejvíce flexibilní z uvedených formátů a umožňuje vybrání informací, které se mají v rámci logu zaznamenávat.

```
#Software: Microsoft Internet Information Server 4.0
#Version: 1.0
#Date: 1998-11-19 22:48:39
#Fields: date time c-ip cs-username s-ip cs-method cs-uri-stem cs-uri-query sc-status sc-bytes cs-
bytes time-taken cs-version cs(User-Agent) cs(Cookie) cs(Referrer)

1998-11-19 22:48:39 206.175.82.5 - 208.201.133.173 GET /global/images/navlineboards.gif - 200 540 324
157 HTTP/1.0 Mozilla/4.0+(compatible;+MSIE+4.01;+Windows+95) USERID=CustomerA;+IMPID=01234
http://yourturn.rollingstone.com/webx?98@@webx1.html
```

Obrázek 3-11 Ukázka W3C Extended Log formátu

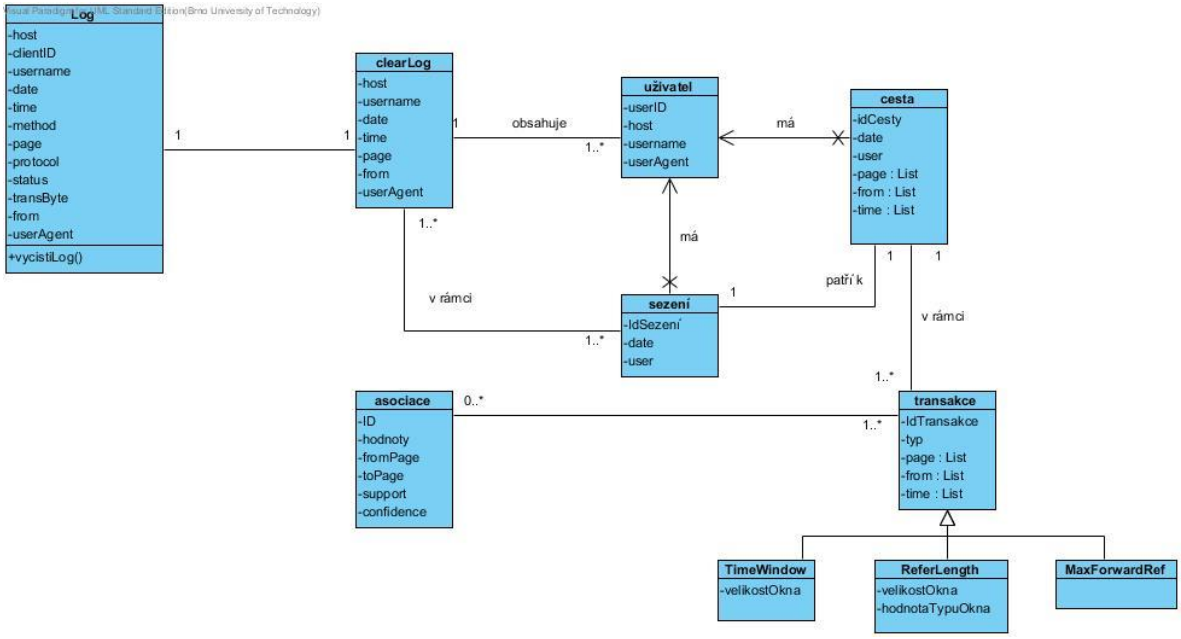
4 Návrh aplikace

Cílem této diplomové práce je navrhnout a vytvořit aplikaci schopnou ze vstupního souboru, obsahující log webového serveru ve formátu NCSA Combined, popsaném v předchozí kapitole, získat asociační pravidla, sloužící k analyzování provozu na zkoumaném Webu. V předchozích kapitolách byly rozebrány nejdůležitější pojmy související s řešenou problematikou. Tato kapitola se bude zabývat návrhem aplikace a jejích funkcí. Aplikace bude vyvíjena jako klasická desktopová aplikace s využitím programovacího jazyka JAVA.

První část kapitoly návrhu se bude zabývat návrhem tříd aplikace, jejich významem a jejich bližším popisem. Druhá část kapitoly bude věnována návrhu a popisu jednotlivých funkcí aplikace. V rámci návrhu čtenář uvidí propojení s pojmy popsanými v předchozích kapitolách, jejichž vykonání je nezbytné pro získání asociačních pravidel, jejichž nalezení je cílem navrhované aplikace.

4.1 Návrh tříd

Cílem je navrhnout třídy takovým způsobem, aby s navrženými třídami bylo možné provádět všechny zamýšlené funkce. Jak znázorňuje obrázek 4-1, hlavní třída je nazvána „Log“. Tato třída obsahuje pole odpovídající informacím získaných ze vstupního souboru. Třída „Log“ tedy bude obsahovat všechny informace a ostatní třídy budou založeny na informacích získaných z této třídy. Třída „Log“ obsahuje i záznamy, které nejsou pro nalezení asociačních pravidel nerelevantní, a proto
je další třídou třída „clearLog“. Tato třída je získána z třídy „Log“ odstraněním záznamů nerelevantních pro nalezení asociačních pravidel. Třída „clearLog“ je tedy výstupem fáze čištění dat popsané v kapitole 3.2.1. Z třídy „clearLog“ jsou následně odvozeny další dvě třídy. Těmito třídami



Obrázek 4-1Díagram tříd

jsou třídy „uživatel“ a „sezení“. Třída „uživatel“ obsahuje seznam unikátních uživatelů, vyskytujících se ve zkoumaném webovém logu. Třída „uživatel“ a v ní obsažený seznam unikátních uživatelů je výstupem fáze identifikace uživatele popsané v kapitole 3.2.2. Každý nalezený uživatel provedl ve zkoumaném logu minimálně jedno sezení, které odpovídá prohlížení webových stránek na serveru, ze kterého log pochází. Jednotlivá sezení jsou uložena ve třídě „sezení“, která získává informace o stránkách prohlížených v rámci sezení z třídy „clearLog“ a každé sezení je spojeno s konkrétním uživatelem (viz. Obrázek 4-1). Získaná sezení jsou výstupem fáze identifikace sezení, která je blíže popsána v kapitole 3.2.3. Z předchozích kapitol také víme, že nalezení sezení je pouze mezikrokem k nalezení asociačních pravidel. Dalším důležitým krokem je z nalezených sezení vytvořit tzv. cesty uživatele. Tedy seřadit stránky v jednotlivých sezeních podle pořadí, v jakém byly uživatelem prohlíženy a doplnění stránek, které v logu nejsou zaznamenány, např. z důvodu použití tlačítka zpět. Pro uchovávání cest uživatele, slouží třída „cesta“, která je svázána s konkrétním uživatelem a sezením. Informace uložené v této třídě jsou výstupem fáze kompletace cesty, která je popsána v kapitole 3.2.4.

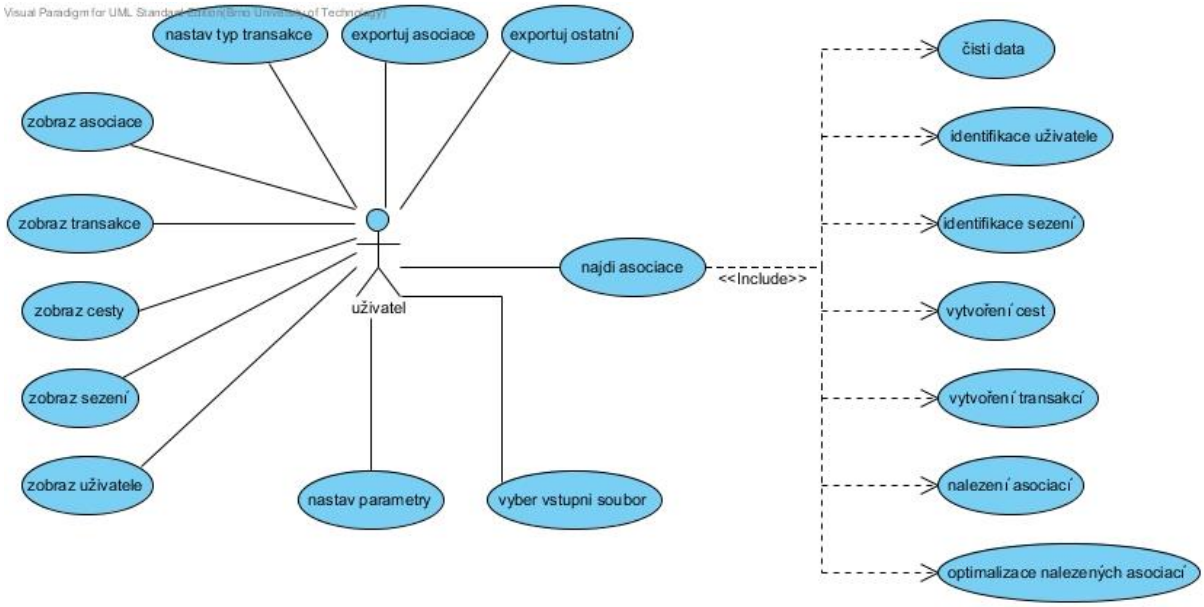
Dosud popsané třídy spadají svým obsahem do fáze předzpracování dat. Předposledním krokem je z uživatelských cest vytvořit transakce, které budou vstupem pro fázi nalezení asociačních pravidel. Transakce budou uloženy v rámci třídy „transakce“. Obrázek 4-1 ukazuje, že třída „transakce“ je propojena s třídou „cesta“, která je výstupem fáze předzpracování dat. Aplikace je dále navržena tak, aby se neomezovala pouze na jeden způsob nalezení transakcí, a proto můžeme na obrázku 4-1 vidět tři třídy, které jsou generalizací třídy „transakce“. Tyto třídy jsou nazvány „TimeWindow“, „ReferLength“ a „MaxForwardRef“, a reprezentují tři přístupy pro vytvoření transakcí z uživatelských cest. Posledním krokem je nalezení asociačních pravidel. Pro uchování informací o nalezených asociačních pravidlech je navržena třída „asociace“. Třída zaznamenává pro každé nalezené asociační pravidlo informaci udávající hodnotu parametrů support a confidence, jejichž význam je blíže popsaný v kapitole 3.3.3 zabývající se asociačními pravidly.

4.2 Návrh funkcí

Cílem návrhu funkcí bylo navrhnout funkce aplikace takovým způsobem, aby umožňovaly uživateli nejen nalezení asociačních, ale také analýzu zjištěných informací. Navržené funkce lze rozdělit do tří kategorií. První kategorií jsou funkce sloužící pro nastavení parametrů vyhledávání a funkce pro samotné vyhledávání. Do druhé kategorie se řadí funkce pro analýzu zjištěných informací a poslední kategorie obsahuje funkce pro export informací.

Do první kategorie lze zařadit funkce nastav parametry, vyber vstupní soubor, nastav typ transakce, najdi asociace (viz. Obrázek 4-2). Funkce „vyber vstupní soubor“ jak už její název napovídá, slouží k vybrání souboru, obsahujícího webový log, který se má zpracovat. Zajímavější je funkce „nastav parametry“. Tato funkce slouží k nastavení parametrů, na jejichž základě se bude

provádět fáze předzpracování dat i samotné nalezení asociačních pravidel. Funkce bude uživateli umožňovat nastavit, který typ souborů je pro analýzu důležitý, např. pokud se bude zpracovávat log serveru zabývající se sdílením fotek nebo videí, tak bude zcela jistě nežádoucí vynechat tyto typy souborů z analýzy. Dalším z nastavitelných parametrů budou hodnoty support a confidence, které ovlivňují nalezené asociační pravidla. Další funkce z této kategorie je „nastav typ transakce“, která slouží k nastavení způsobu rozlišení transakcí. Jak bylo uvedeno u návrhu tříd, návrh předpokládá tři způsoby nalezení transakcí. Tyto způsoby jsou blíže popsány v kapitole 3.3.3. Funkce nastavení typu transakce je úzce propojená s funkcí nastavení parametrů, a to z toho důvodu, že volba typu transakce sebou nese i nastavení potřebných atributů. Např. u metody Time window musí uživatel nastavit hodnotu, udávající délku trvání časového okna a u metody Reference length je nutné nastavit kromě hodnoty délky časového okna také hodnotu, sloužící pro rozlišení, zda se jedná o navigační nebo obsahovou stránku. Poslední funkcí je funkce „najdi asociace“, sloužící pro nalezení asociačních



Obrázek 4-2 Diagram případů užití

pravidel. Jak můžeme vidět na obrázku 4-2 tato funkce vyžaduje pro svoji funkčnost několik dalších funkcí, které zajišťují provedení předzpracování dat, nalezení transakcí, asociací a optimalizaci nalezených asociací. Jedná se o „pomocné“ funkce, které nejsou používány přímo uživatelem, ale jsou nezbytné pro správnou funkčnost uživatelem používaných funkcí.

Druhou kategorii tvoří funkce podporující analýzu nalezených informací. Do této kategorie náleží funkce zobraz asociace, zobraz transakce, zobraz cesty, zobraz sezení, zobraz uživatele. Primární funkcí pro analýzu je funkce zobraz asociace, která zobrazí nalezené asociační pravidla. Pro analýzu však mohou být užitečné i další informace získané v procesu nalezení asociačních pravidel. Proto jsou součástí návrhu i funkce pro zobrazení všech získaných informací. Jednotlivé funkce z velké části odpovídají krokům předzpracování dat. Funkce pro zobrazení sezení, cest a transakcí se

jeví jako vhodné pro implementace různých filtrů. Např. filtry zobrazení sezení pouze jednoho konkrétního uživatele, nebo zobrazení sezení provedených v konkrétní datum, intervalu, atd.

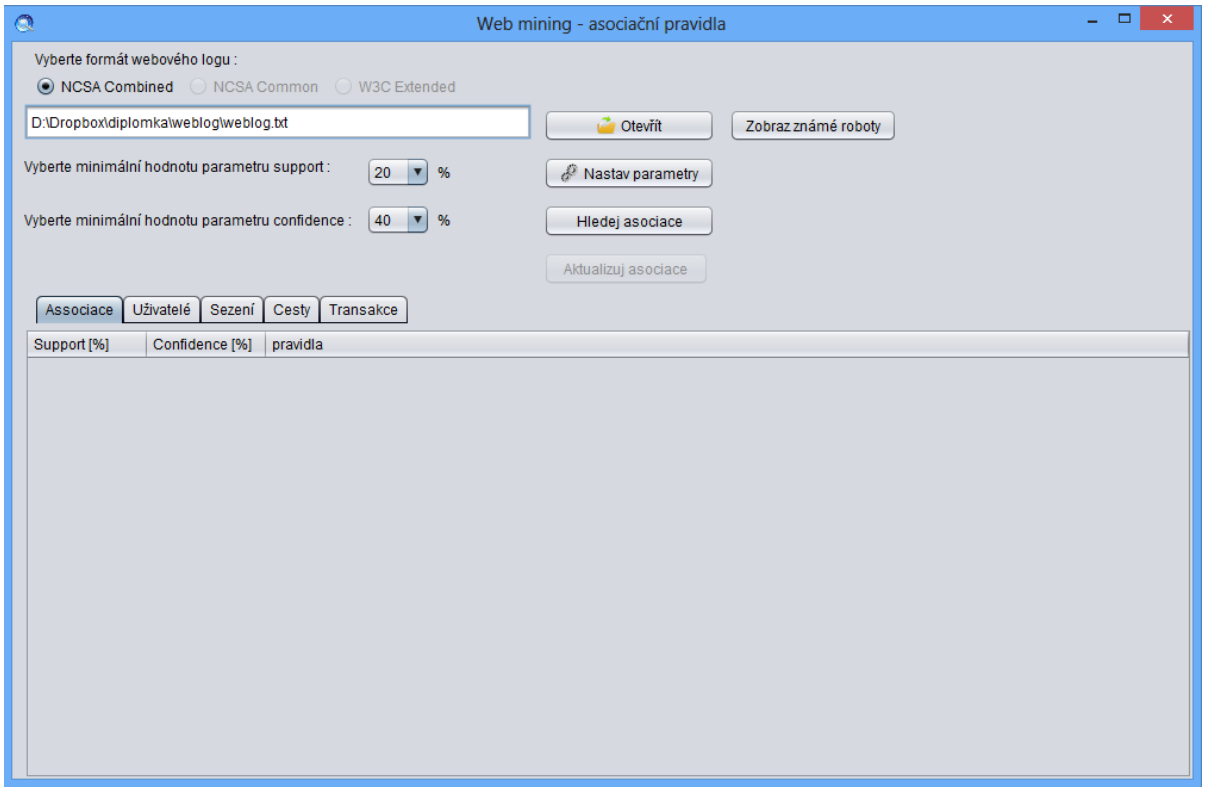
Poslední kategorií jsou funkce pro export informací. Zde se opět primárně jedná o export nalezených asociačních pravidel, ale stejně jako u předchozí kategorie, i zde aplikace v návrhu počítá s exportem všech ostatních informací. Návrh počítá primárně s exportem do formátu „pdf“, ale v konečné implementaci se může objevit i export do některého textového formátu, např. formátu „doc“.

5 Implementace aplikace

Jak již bylo v předchozích kapitolách zmíněno, aplikace je implementována v programovacím jazyce Java a jedná se tedy o desktopovou aplikaci. Předchozí kapitoly byly věnovány definování pojmu získávání znalostí z Webu s následným zaměřením a detailním vysvětlením pojmu získávání znalostí z užití Webu, kterým se tato diplomová práce zabývá. V aktuální kapitole se zaměříme na způsob, jakým jsou jednotlivé kroky získávání znalostí z užití Webu implementovány. Základem pro implementaci aplikace byl návrh prezentovaný v kapitole 4. První část kapitoly bude věnována seznámení se s hlavním oknem aplikace a možnostmi, které aplikace nabízí. Ve druhé části bude prezentován způsob implementace jednotlivých funkcí aplikace. V rámci implementace aplikace bylo využito i několik volně dostupných knihoven pro jazyk Java. Důvod použití těchto knihoven bude vysvětlen v rámci popisu implementace funkcí aplikace.

5.1 Popis aplikace

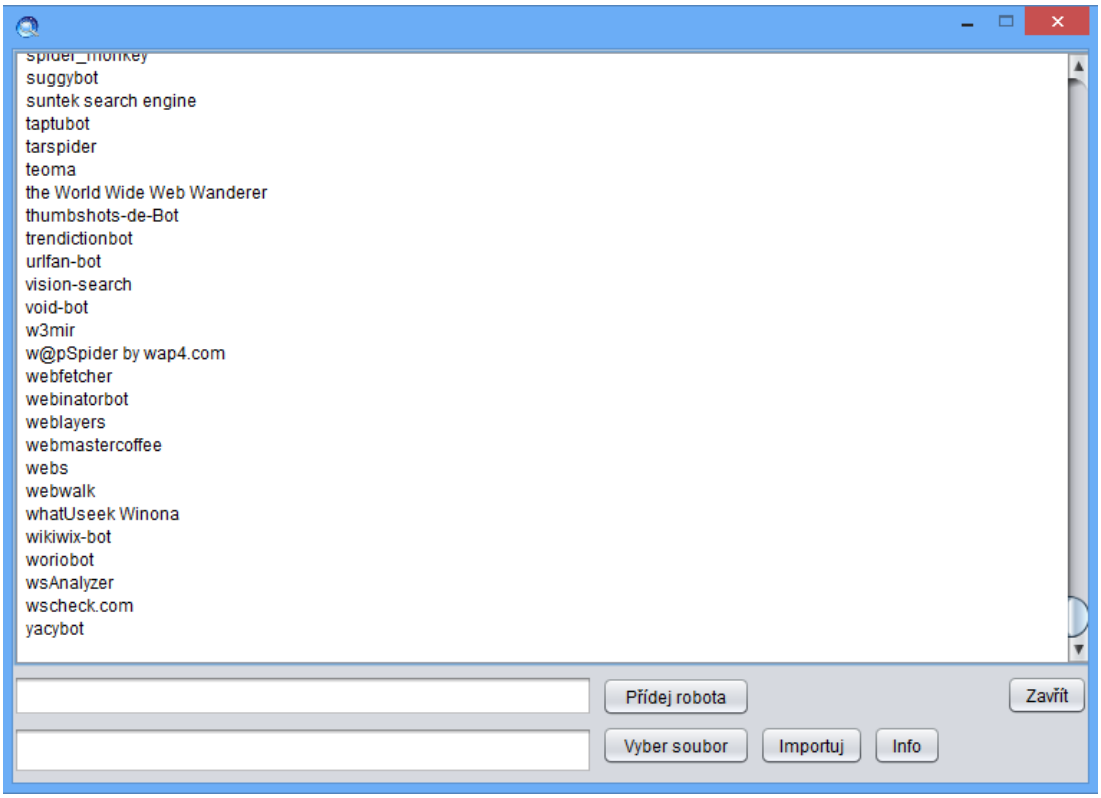
Hlavní okno aplikace lze z pohledu funkčnosti rozdělit na dvě části. Část obsahující ovládací prvky a část obsahující prvky pro zobrazení získaných informací. Ovládací část se skládá z několika tlačítek, textového pole, radiobuttony (přepínací tlačítka) a dvěma comboboxy (rozbalovací seznam). Přepínací tlačítka slouží pro výběr formátu webového logu, který se bude analyzovat. Jak bylo



Obrázek 5-1 Hlavní okno aplikace

zmíněno v předchozích kapitolách, vytvářená aplikace pracuje s NCSA combined formátem webového logu. Z tohoto důvodu, jak můžeme vidět na obrázku 5-1, jsou ostatní přepínací tlačítka v současnosti neaktivní. Protože se však možnost rozšířit aplikaci o možnost pracovat i s ostatními formáty webového logu jeví jako vhodné případné rozšíření aplikace, je možnost volby vložena do aplikace již nyní. Dalšími ovládacími prvky jsou dva rozbalovací seznamy, které slouží k nastavení parametrů support a confidence. Tyto parametry ovlivňují vyhledávání asociačních pravidel, jak bylo uvedeno v 3.3.3. Předposledním prvkem ovládací části je textové pole, do kterého se zobrazuje cesta k analyzovanému souboru. Posledním prvkem je soubor 5 tlačítek, určených k ovládání aplikace. Tlačítko „otevřít“ slouží k výběru webového logu, který se má analyzovat. Po dokončení výběru se cesta k souboru vypíše do dříve popsaného textového pole. Dalším je tlačítko „Nastav parametry“. Toto tlačítko slouží k zobrazení dalšího okna, které poskytuje rozšiřující možnosti nastavení aplikace. Okno s nastavením se skládá z možnosti nastavení typu transakce, způsobu určení sezení a volbě formátů souborů, které se mají do analýzy zahrnout. Aplikace poskytuje dva způsoby určení transakcí a tři způsoby pro určení sezení. Pro identifikaci transakcí jsou implementovány metody time window a maximal forward reference. V případě identifikace sezení se jedná o metody založené na maximální délce sezení, odstupu mezi odkazy a metodě založené na adrese URL v poli referrer webového logu. Všechny uvedené metody byly popsány v kapitole 3. Třetím tlačítkem je „Hledej asociace“, které provádí vyhledání asociačních pravidel. Protože slouží k vyhledávání asociací, vykonává i všechny ostatní kroky potřebné k nalezení asociačních pravidel tak, jak byly popsány v kapitole 3. Předposledním tlačítkem je „Aktualizuj asociace“, jež slouží k novému vyhledání asociací. Na rozdíl od předchozího tlačítka však neprovádí jednotlivé kroky fáze předzpracování dat, ani určení transakcí. Provádí tedy pouze novou analýzu z již získaných dat. Jeho použití je především při změně parametrů support a confidence, kdy není nutné znovu provádět fázi předzpracování dat. Důvodem je, že pokud bychom při každé změně parametrů support a confidence prováděli kompletní analýzu, tak při velkém vstupním webovém logu by analýza byla náročná a navíc i zbytečná protože data o nalezených uživatelích, jejich sezeních, cestách a transakcích se stejně nemění. Tyto data by se změnily pouze při změně parametrů, přístupných přes tlačítko „Nastav parametry“. Posledním tlačítkem je „Zobraz známé roboty“. Toto tlačítko zobrazí nové okno, které obsahuje seznam známých robotů. Zobrazené okno obsahuje i možnost přidat robota, který nebyl rozpoznán v kroku čištění dat, do seznamu známých robotů. Bližší význam tohoto seznamu bude vysvětlen později. Okno zobrazující známé roboty můžeme vidět na obrázku 5-2.

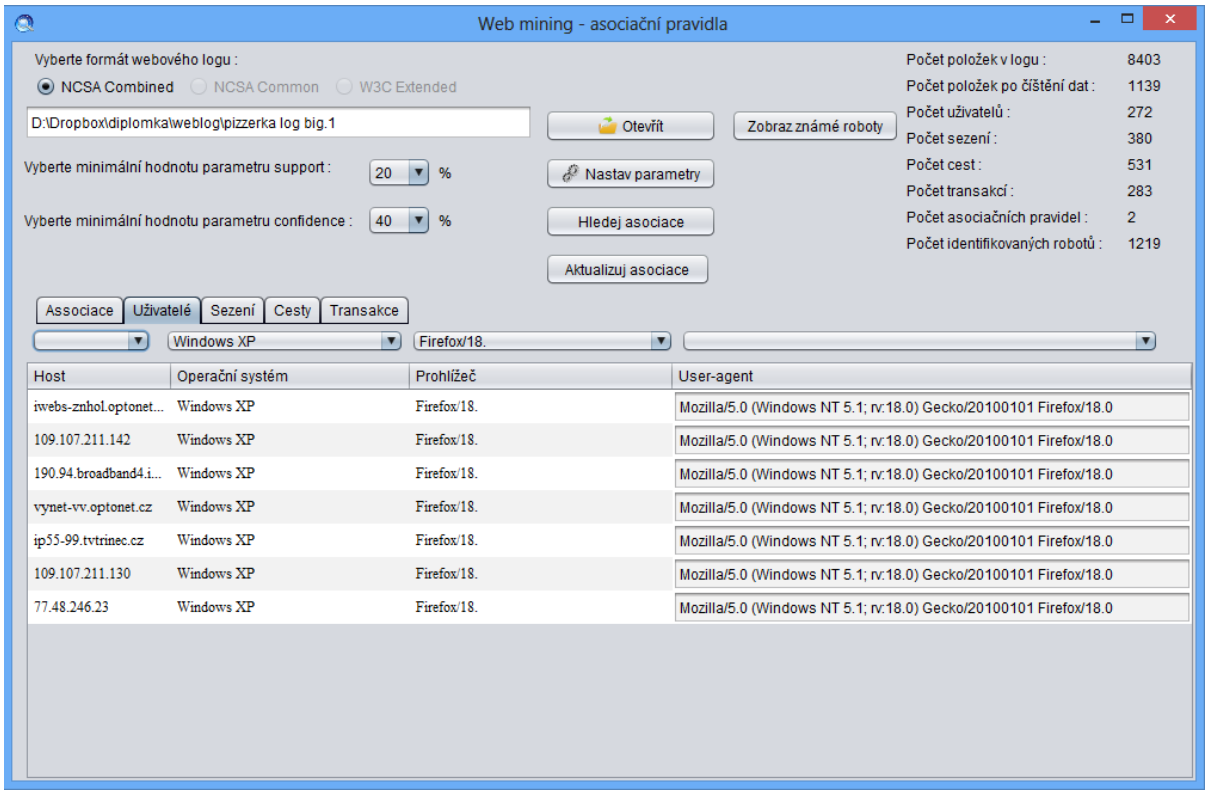
Část zobrazující informace získané analýzou se skládá z jednoho panelu. Tento panel obsahuje pět záložek, které reprezentují a zobrazují informace získané po provedení jednotlivých kroků úlohy získávání znalostí z webového logu. Jak můžeme vidět na obrázku 5-1 jsou jednotlivé záložky pojmenovány „Asociace“, „Uživatelé“, „Sezení“, „Cesty“ a „Transakce“. Obsah každé záložky je tvořen tabulkou, do které se zobrazují příslušné informace. Počet sloupců v jednotlivých tabulkách se liší a je závislý na počtu informací, které je nutné v příslušné tabulce zobrazit. Blíže



Obrázek 5-2 Okno zobrazující známé roboty

budou tabulky, jedna po druhé, představeny v části kapitoly zabývající se funkcemi aplikace. Kromě tabulky sloužící k zobrazení informací obsahuje každá záložka také pop-up menu, které poskytují některé další funkce. Těmito funkcemi jsou například export získaných dat do formátu pdf, zobrazení grafů vytvořených ze získaných dat, zobrazení detailu vybraného řádku tabulky atd. Podobně jako u počtu sloupců u tabulek, se i nabídky pop-up menu liší u každé záložky a budou popsány blíže v části kapitoly zabývající se popisem funkcí aplikace. Jelikož je cílem aplikace provádět analýzu, a protože mohou data získaná z webového logu obsahovat velké množství informací, bylo nezbytné implementovat možnost filtrovat záznamy. Tato skutečnost tedy vedla k tomu, že pro každou tabulku, kromě tabulky „asociace“, a pro každý její sloupec byl vytvořen filtr umožňující filtrovat záznamy v tabulce dle zvoleného sloupce. Navíc, jelikož možnost filtrovat pouze pomocí jediného sloupce nemusí být dostatečná, byly filtry u jednotlivých tabulek implementovány tak, že je možné filtry u jednotlivých sloupců tabulky kombinovat. Díky tomu je možné mnohem lépe analyzovat záznamy v tabulkách, protože lze tímto způsobem odfiltrovat velké množství informací. Filtry jsou implementovány formou rozbalovacího seznamu (tj. comboBoxu). Tyto seznamy obsahují vždy všechny položky, které se v tabulce vyskytují v příslušném sloupci. Tak je zaručeno, že bude možno filtrovat všechny položky nacházející se v příslušné tabulce. Práci s filtry tabulky je možné vidět na obrázku 5-3.

Účelem této kapitoly bylo seznámit se s rozhraním aplikace a krátkým nastíněním jejich možností. V dalších částech se zaměříme na popis funkčnosti jednotlivých funkcí aplikace, z nichž některé již byly zmíněny, jiné budou v následujících částech odhaleny.



Obrázek 5-3 Práce s filtry tabulky

5.2 Funkce aplikace

V této části kapitoly si představíme všechny funkce aplikace spolu s popisem jejich funkčnosti a způsobu implementace. Jako první budou zmíněny funkce, které jsou hlavní náplní aplikace. Tedy všechny kroky předzpracování dat popsaných v kapitole 3.2, a také nalezení asociačních pravidel. Funkce implementující předzpracování dat jsou vytvořeny pro formát NCSA combined webového logu. Každá z těchto funkcí je implementována v samostatné třídě, která implementuje definované rozhraní. Rozšíření aplikace o možnost zpracování dalších formátů logů je tedy možná doplněním tříd, které budou implementovat definovaná rozhraní pro daný formát webového logu. Následně budou popsány ostatní funkce, které byly implementovány pro zajištění větší podpory při analyzování získaných informací.

5.2.1 Čištění dat

Čištění dat je jednou z nejdůležitějších funkcí aplikace. Důvodů, proč se jedná o velice důležitou funkci, je hned několik. Tato funkce nejenom reguluje velké množství dat načtených ze zkoumaného logu, ale do značné míry také ovlivňuje správnost výstupů ostatních funkcí. Důvodem je, že právě v rámci této funkce se určují data, se kterými se bude dále pracovat. A tedy pokud by byla regulace dat provedena nesprávně, budou ostatní výsledky zkresleny a nebudou reprezentovat skutečný stav. A právě proto bylo nutné věnovat implementaci této funkce velkou pozornost.

V aplikaci, vytvářené v rámci této diplomové práce, je čištění dat implementováno následovně. První regulace dat je založena na stavovém kódu záznamu v logu. Tato regulace vychází ze skutečnosti, že v procesu analýzy nás zajímají pouze úspěšně dokončené požadavky. Proto bylo prvním krokem projít záznamy načtené z logu a smazat všechny záznamy, jejichž stavový kód není 200. Tímto krokem jsme získali pouze úspěšné přístupy na zkoumaný Web. Protože zkoumáme chování uživatelů na Webu, je další redukce dat založena na metodě http dotazu. Běžní uživatelé používají pouze metody GET a POST, na základě této skutečnosti můžeme provézt další redukci, kdy ze zkoumaných dat odstraníme záznamy, které používají jiné než výše uvedené dotazovací metody. Tento krok může taktéž vézt poměrně k významné redukci zkoumaných dat, nýbrž někteří roboti při procházení Webů používají metodu HEAD, a záznamy obsahující tuto metodu jsou díky uvedené redukci odstraněny.

Předposlední redukce vychází z toho, že ne všechny položky zaznamenané v rámci logu, a načítané při přístupu na webovou stránku, jsou pro analýzu důležité. Řada položek je načítána jako součást stránky, např. obrázky tvořící grafiku stránky, soubor obsahující css styly stránky, skripty v jazyce javascript, atd. Takovéto položky nejsou pro analýzu důležité, ba právě naopak je nutné tyto položky odstranit. Z tohoto důvodu jsou v základním nastavení aplikace odstraněny všechny položky, které neobsahují typ souboru odpovídající příponám webových stránek, jako např. html, php, asp, java, atd. V kapitole 5.1 jsme si však již uvedli, že aplikace poskytuje možnost nastavení. Okno s natavením poskytuje i možnost vybrat přípony souborů, které chceme do analýzy zahrnout. Tímto způsobem lze analyzovat i Web, který např. obsahuje řadu dokumentů ve formátu pdf, a začlenění přístupu na tyto dokumenty je pro člověka provádějícího analýzu důležité.

Posledním krokem je ze zbylých dat odstranit záznamy, které nebyly vytvořeny uživateli, nýbrž roboty procházejícími Web. Jak již bylo zmíněno, aplikace pracuje s formátem webového logu NCSA combined, který obsahuje pole user-agent. Toto pole je využito pro rozpoznávání robotů od běžných uživatelů. Část robotů je v aplikaci filtrována pomocí regulárního výrazu, ale protože ne všichni roboti se dají identifikovat pomocí klíčových slov, jako např. bot, crawler, spider, atd., existuje v aplikaci i druhý způsob filtrování robotů. Druhý způsob je založený souboru, který obsahuje seznam známých robotů vyskytujících se na internetu. Aplikace si načte obsah tohoto souboru a následně provádí smazaní záznamů, které v poli user-agent obsahují některého ze známých robotů. Tento souboru byl vytvořen na základě informací získaných z [10] a [11]. Protože se neustále objevují noví roboti, a tedy seznam nemusí obsahovat všechny existující roboty, poskytuje aplikace možnost přidat do seznamu nového robota, případně importovat do aplikace zcela nový soubor obsahující seznam robotů. Okno aplikace, které poskytuje tyto možnosti, je zobrazeno na obrázku 5-2.

Po provedení všech výše uvedených kroků zůstanou v aplikaci pouze relevantní data pro analýzu, která jsou předána na další zpracování.

5.2.2 Identifikování uživatelů

Cílem je, jak bylo popsáno v kapitole 3.2.2, identifikovat uživatele přistupující na Web. Základem pro identifikaci uživatelů je pole host z webového logu. Toto pole obsahuje IP adresu, případně doménové jméno uživatele. Problémem však je, že v dnešní době často přistupuje více uživatelů na Web ze stejné IP adresy. Tato skutečnost vede k tomu, že použití pouze pole host pro identifikaci uživatele je nedostačující. Proto se v rámci aplikace používá pro identifikaci uživatele kombinace dvou polí webového logu. Těmito poli jsou pole host a user-agent. U každého záznamu získaného z funkce čistění dat jsou porovnány uvedené pole s identifikovanými uživateli, a pokud se tyto pole shodují, jedná se o již identifikovaného uživatele. V opačném případě se jedná a nového uživatele, který je přidán do seznamu identifikovaných. Informace o identifikovaných uživatelích jsou zobrazovány do tabulky „uživatelé“, která se skládá ze čtyř sloupců, jmenovitě Host, Operační systém, Prohlížeč, User-agent (viz.Obrázek 5-3). Údaje pro sloupce operační systém a prohlížeč jsou získány z pole user-agent webového logu. Protože však pole user-agent nemá jednotný formát, je poměrně obtížné tyto údaje získat, jelikož se toto pole liší mezi různými prohlížeči a operačními systémy. Pro získání těchto údajů byly implementovány dvě funkce, které provádějí identifikaci operačního systému a webového prohlížeče uživatele. Rozpoznávání obou údajů je založeno na informacích získaných z [10]. Přestože je aplikace schopná rozpoznat přes 160 různých prohlížečů a 60 operačních systému, lze i přesto nalézt operační systémy a prohlížeče, které aplikace nerozpozná. Počet nerozpoznaných systémů a prohlížečů je však poměrně nízký. Skutečnost že je aplikace schopná rozpoznat velké množství prohlížečů a systému, je užitečná pro tvorbu grafů, které zobrazují statistiky použitých operačních systémů a webových prohlížečů. O těchto grafech bude blíže pojednáno v dalších kapitolách.

Po provedení identifikace uživatelů předá aplikace údaje o identifikovaných uživatelích a všech záznamech, které daný uživatel vytvořil v logu, funkci pro identifikaci sezení. Čímž je ukončena funkce identifikování uživatelů.

5.2.3 Identifikování sezení

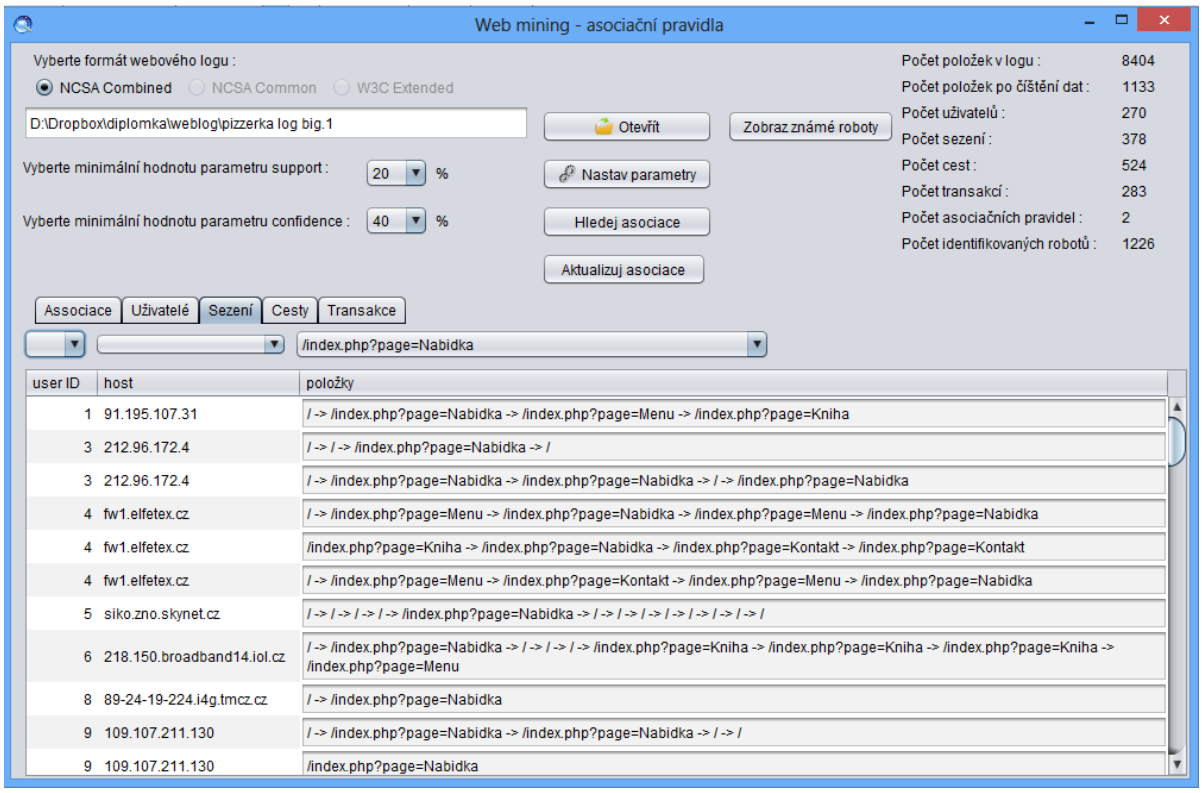
Funkce pro identifikování sezení využívá jako vstup údaje získané identifikací uživatelů. U každého uživatele je totiž uložený seznam položek, které daný uživatel při návštěvách Webu vytvořil v logu. Proto tato funkce postupně prochází všechny identifikované uživatele a vytváří uskutečněná sezení. Jak bylo napsáno v kapitole 3.2.3, existuje více způsobů, jak určit sezení. Ve výchozím nastavení určuje aplikace sezení na základě časové metody, kdy doba sezení nesmí přesáhnout délku 30 minut. Podobně jako u funkce čištění dat je však možné toto nastavení změnit. Aplikace umožňuje vybrat způsob určení sezení ze tří možností. Jedná se o metody popsané v kapitole 3.2.3. U druhé časové metody určující sezení na základě max. doby prohlížení stránky, je hodnota prahu stanovena na 15 minut. Poslední, strukturně orientovanou, metodu bylo možné implementovat díky volbě formátu

webového logu, se kterým aplikace pracuje a který umožňuje provedení této metody. U časových metod byla z důvodu nutnosti určení rozdílu mezi jednotlivými záznamy použita knihovna Jodatime. Nalezená sezení jsou podobně jako uživatelé zobrazovány do tabulky. Tabulka „sezení“ se skládá ze tří sloupců, kterými jsou user ID, host a položky. Jedná se o pole udávající identifikaci uživatele, jeho IP adresu, resp. doménové jméno a položky (stránky) navštívené v rámci sezení. Obrázek 5-4 ukazuje tabulku sezení s aktivním filtrem, který vybírá všechna sezení, v nichž bylo přistoupeno na stránku zvolenou filtrem. Je vhodné dodat, že možnost volby typu určení sezení může vézt ke skutečnosti, že nalezená sezení nemusí být ve všech případech shodné. A tato skutečnost se taktéž může promítnout do výsledku nalezení asociačních pravidel.

Po provedení identifikace sezení jsou získané údaje předány další funkci. Touto funkcí je identifikace cest, která bude dále pokračovat v provádění předzpracování dat.

5.2.4 Identifikování cest

Dalším krokem v procesu nalezení asociačních pravidel je identifikování cest, nebo přesněji dokončení mapování cesty uživatele v rámci sezení. Jak bylo uvedeno v kapitole 3.2.4, je nutné tuto úlohu provádět, protože ne všechny přístupy na stránky jsou uloženy v logu. V aplikaci je identifikování cest řešeno prostřednictvím pole referrer webového logu. Pro každé sezení, získané pomocí funkce identifikování sezení, se postupně prochází všechny položky náležící do daného sezení a kontroluje se, zda se shoduje obsah v poli referrer s obsahem v poli request předchozí položky v sezení. Porovnává se tedy URL stránky, ze které se na aktuální stránku přistoupilo, a URL stránky požadované v předchozím dotazu. Pokud je shoda nalezena, je položka zařazena do cesty a obsah cesty zůstává stejný jako obsah sezení. V opačném případě se zpětně projdou všechny položky zařazené do aktuální cesty, a podobně jako v předchozím případě se hledá shoda. Ve chvíli, kdy je shoda nalezena, jsou do cesty přidány záznamy o všech URL, přes které je nutné se vrátit, abychom se dostali na stránku, ze které bylo na aktuální zkoumanou URL přistoupeno. Pokud shoda není ani v tomto případě nalezena, je položka zařazena do aktuální cesty bez přidávání zpětných odkazů. Tato skutečnost může nastat např. v případě, že je zkoumaná stránka a stránka z pole referrer zařazena do rozdílných sezení.



Obrázek 5-4 Tabulka sezení s aktivním filtrem

Získané údaje o cestách jsou, stejně jako v přechozích případech, zobrazována do tabulky. Tabulka zobrazující informace o cestách se skládá ze tří sloupců. Prvním sloupcem je sloupec user ID, sloužící pro identifikaci uživatele, který cestu provedl. Druhým sloupcem je identifikace sezení. Třetí, a poslední sloupec, udává cestu uživatele, tedy jedná se o seznam URL udávající pohyb uživatele po Webu. Tabulka zobrazující identifikované cesty vypadá obdobně jako tabulka zobrazující sezení. Tabulku zobrazující sezení jsme mohli vidět na obrázku 5-4. Na uvedených obrázcích je možné si povšimnout, že např. první sezení a první cesta se liší. Tento rozdíl je výsledkem kompletace cesty. Získané výsledky jsou předány funkci provádějící identifikaci transakcí, která provádí poslední úpravu dat před aplikací algoritmu pro nalezení asociačních pravidel.

5.2.5 Identifikování transakcí

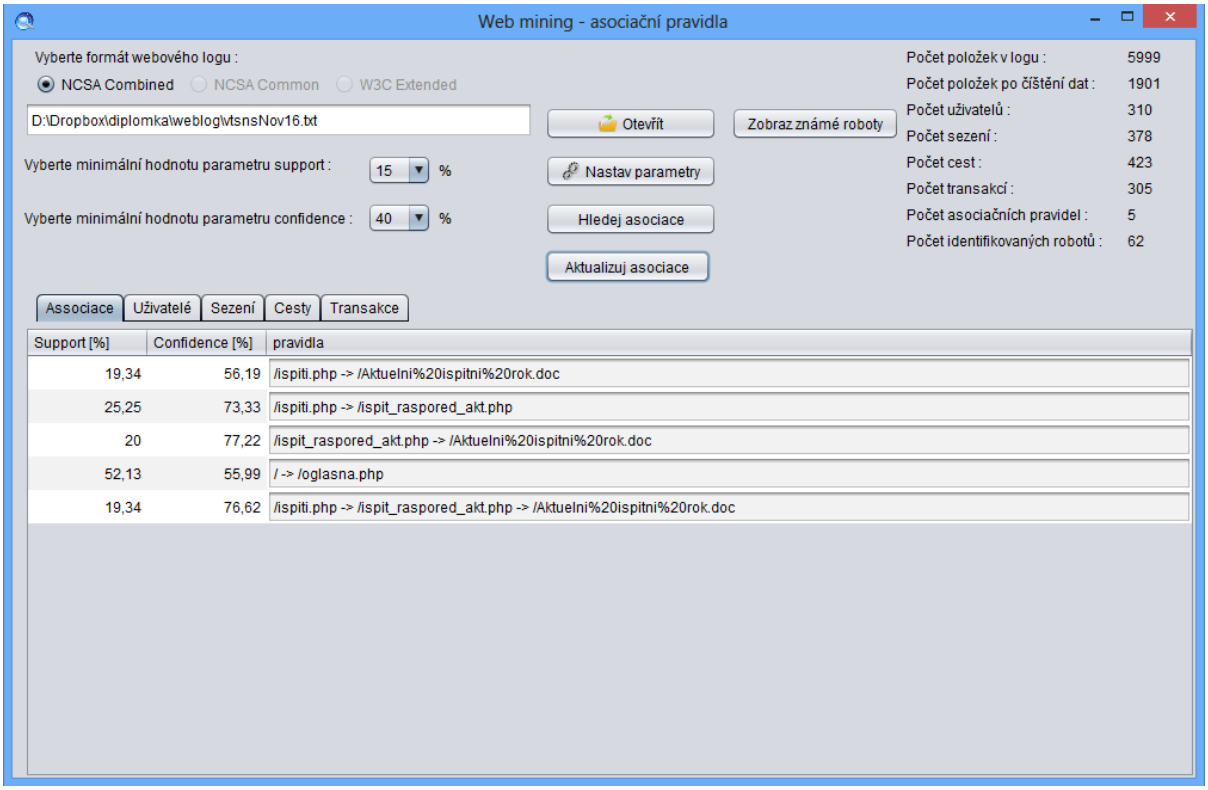
Předposledním krokem procesu získávání znalostí z webového logu je vytvoření transakcí z identifikovaných cest. Jak bylo uvedeno v kapitole 3.3.3, existuje více možností, jak tyto transakce vytvořit. Aplikace poskytuje možnost identifikovat transakce dvěma způsoby uvedenými ve zmíněné kapitole. Jedná se o metody časového okna a maximal forward reference. Ve výchozím nastavení aplikace je pro identifikaci transakcí použita metoda časového okna s délkou okna nastavenou na 30 minut. Velikost časového okna i metodu identifikace transakcí je možné nastavit. Identifikace transakcí je postupně aplikována na všechny cesty získané předchozí funkcí. Po provedení identifikace transakcí je následně provedena redukce, při které jsou odstraněny všechny transakce, které obsahují pouze jedinou položku. Tento krok je proveden z důvodu, že tyto transakce jsou pro

nalezení asociačních pravidel nepotřebné. V případě velkého počtu výskytů takovýchto transakcí mohou hledání asociačních pravidel i negativně ovlivnit. A z těchto důvodů jsou z dalšího zpracování odstraněny.

Získané transakce jsou opět zobrazovány do tabulky. Tabulka je tentokrát tvořena čtyřmi sloupci. První sloupec obsahuje identifikaci uživatele, druhý sloupec IP adresu uživatele, třetí identifikaci cesty a čtvrtý sloupce obsahuje seznam URL adres v transakci. Informace o nalezených transakcích jsou předána funkci pro nalezení asociačních pravidel.

5.2.6 Nalezení asociačních pravidel

Jedná se o funkci, jejímž výstupem je seznam asociačních pravidel, jejichž nalezení bylo hlavním cílem aplikace. Nalezení asociačních pravidel je v aplikaci implementováno algoritmem Apriori. Tento algoritmus byl popsán v kapitole 3.3.3. Algoritmus využívá při vyhledávání asociačních pravidel dva parametry, support a confidence. Oba tyto parametry je možné v aplikaci nastavit pomocí rozbalovacích seznamů. Volba těchto parametrů do značné míry ovlivňuje počet nalezených asociačních pravidel. Špatná volba parametrů může vézt k situaci, kdy nebude nalezeno žádné



Obrázek 5-5 Tabulka zobrazující nalezená asociační pravidla

asociační pravidlo. Algoritmus funguje tak, že v každém kroku vygeneruje množinu n-tic, kde n udává číslo kroku, a následně se vyhodnocuje, které n-tice jsou tzv. frequent (tzn. časté), neboli které n-tice splňují zvolené parametry. Množiny n-tic jsou generovány ze stránek vyskytujících se v nalezených transakcích. Množiny n-tic jsou navíc v každém kroku generovány pouze z frequent množin velikost n-1. Algoritmus ukončí svoji činnost v okamžiku, kdy je frequent množina prázdná

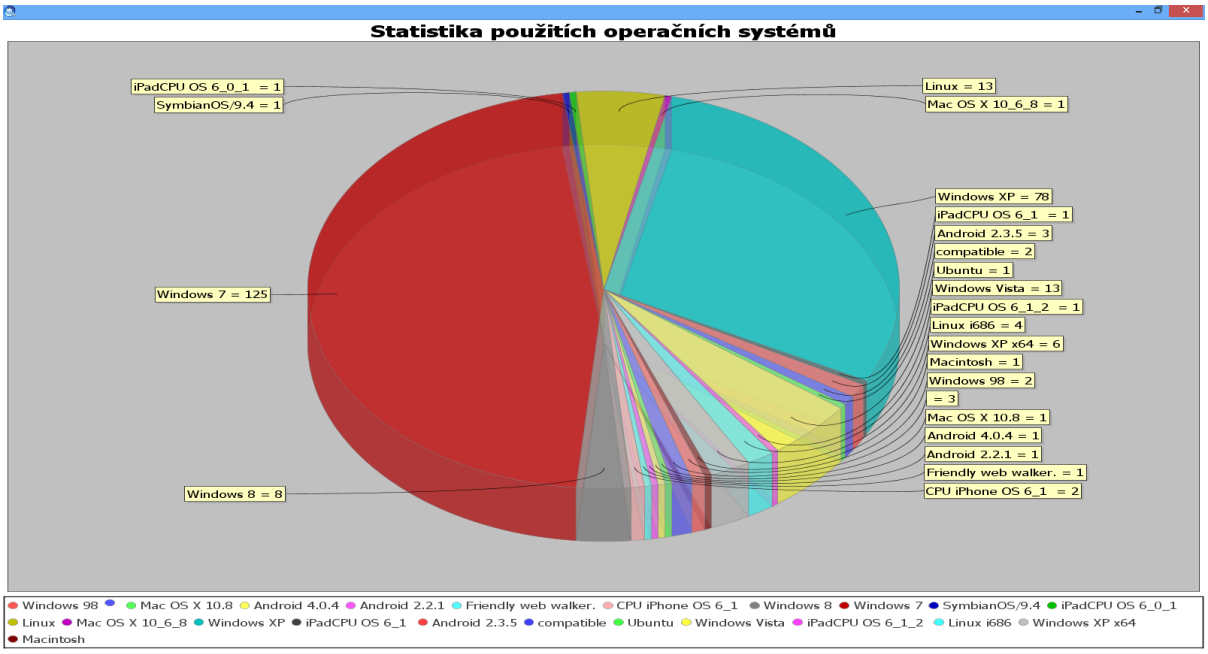
Po ukončení algoritmu jsou nalezená asociační pravidla vypsána do tabulky. Tabulka je tvořena třemi sloupci, kde první sloupec udává hodnotu support daného pravidla v procentech. Druhý sloupec udává hodnotu confidence pravidla v procentech a poslední sloupec obsahuje seznam stránek, které tvoří dané pravidlo. Nalezená asociační pravidla jsou nejčastěji tvořena dvojicemi, případně trojicemi, ale počet stránek v pravidlu není algoritmem, ani jeho implementací v této aplikaci nijak omezen. Obrázek 5-5 ukazuje nalezená asociační pravidla. Můžeme vidět, že aplikace nalezla při daném nastavení parametrů 5 pravidel, z nichž 4 pravidla jsou velikosti 2, a jedno pravidlo je velikosti 3.

5.2.7 Ostatní funkce

V několika předchozích podkapitolách byly popsány hlavní funkce aplikace. Tato kapitola se bude zabývat popisem zbývajících funkcí aplikace, které dosud nebyly představeny. Jak bylo uvedeno v kapitole 5.1, je nad každou tabulkou zobrazující získané informace vytvořeno pop-up menu. Právě tyto pop-up menu zpřístupňují zbylé funkce, které budou popsány v této kapitole.

Jako první začneme u tabulky uživatelé. Pop-up menu této tabulky obsahuje celkem 5 položek. První položka menu zajišťuje export informací o nalezených uživatelích do dokumentu ve formátu pdf. Exportovaná data mohou sloužit např. pro pozdější analýzu. Pro zajištění funkce exportování dat do formátu pdf byla použita knihovna itext. Jde o open source knihovnu dostupnou na Webu. Druhá položka slouží k zobrazení grafu udávajícího statistiku operačních systémů používaných uživateli. Z grafu lze snadno získat přehled o tom, který operační systém návštěvníci nejčastěji používají. Ukázku tohoto grafu je možné vidět na obrázku 5-6. K vytvoření tohoto grafu, stejně jako i k vytvoření ostatních grafů, které budou představeny v následujícím textu, byla použita knihovna jfreechart. Tato knihovna je volně dostupná na Webu. Další položka, třetí v pořadí, vytváří graf zobrazující statistiku webových prohlížečů uživatelů. Obdobně jako u předchozího grafu lze i zde snadno získat přehled prohlížečích, které uživatelé nejčastěji používají. Získané informace mohou být užitečné např. při rozhodovaní o vytvoření verze zkoumaného webu pro mobilní zařízení. Předposlední položka se vytváří stejně jako předchozí graf prohlížečů. Jde o zjednodušený graf, kde u hlavních prohlížečů, kterými jsou Firefox, Chrome, Internet Explorer, Opera a Safari, nejsou rozlišeny verze. Možnost zobrazení tohoto grafu byla přidána z důvodu, že většina hlavních prohlížečů používá systém častých aktualizací. Z tohoto důvodu existuje velké množství verzí těchto prohlížečů, což může vézt k situaci, že graf prohlížečů bude méně přehledný. Tento zjednodušený graf tuto situaci vhodně řeší. Poslední položka zobrazuje taktéž graf. Jedná se o graf, který pro zvoleného uživatele zobrazí statistiky udávájící, které stránky uživatel navštívil a kolikrát dané stránky navštívil.

Druhou tabulkou je tabulka sezení. U této tabulky je pop-up menu tvořeno 3 položkami. První položka slouží stejně jako v případě předchozí tabulky pro export dat z tabulky do dokumentu ve



Obrázek 5-6 Graf operačních systémů

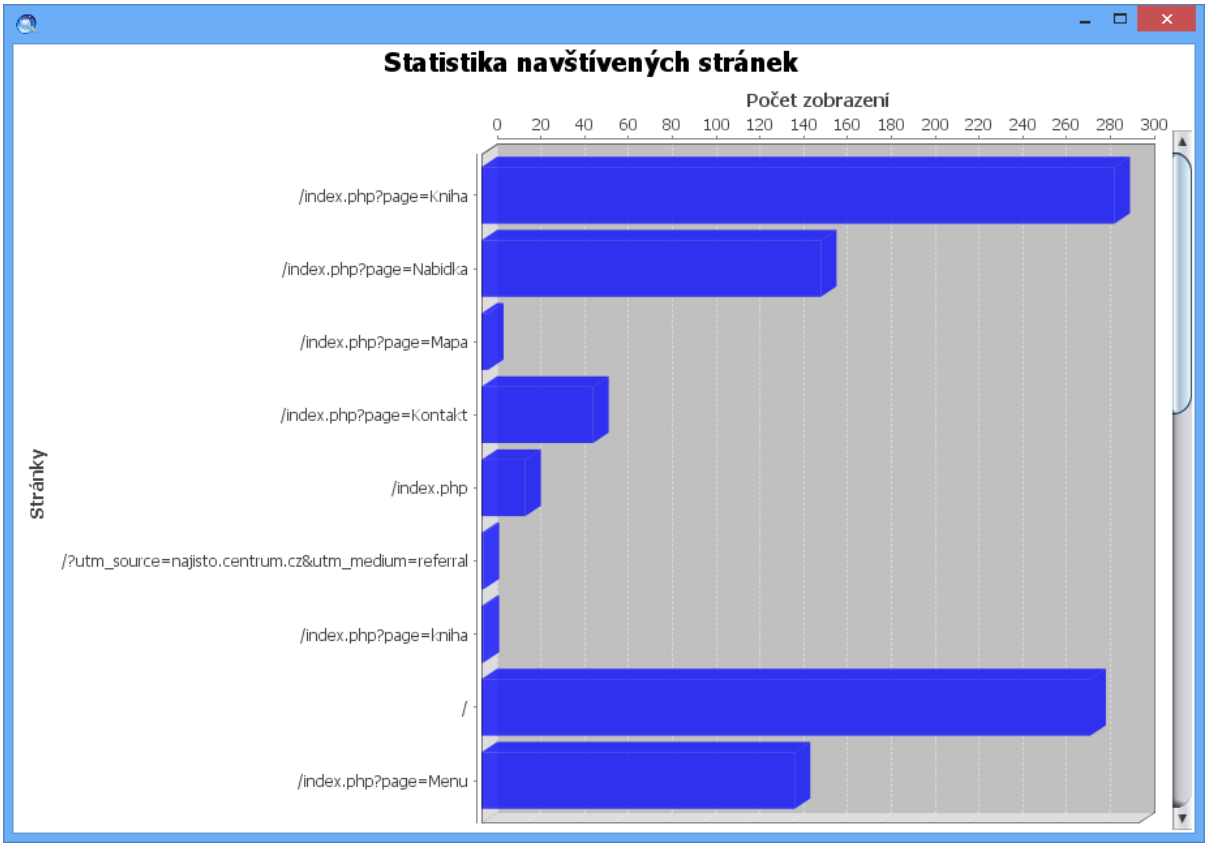
formátu pdf. Druhá položka zobrazí okno obsahující detailnější informace o zvoleném záznamu tabulky. Přesněji okno obsahuje informace o uživateli, operačním systému a prohlížeči. Dále pro každou stránku ve zvoleném sezení udává, ze které stránky bylo na danou stránku přistoupeno a datum a čas přístupu. Poslední položka vytváří okno s grafem, který zobrazuje statistiku všech stránek, na které bylo ve zkoumaném logu přistoupeno a počet přístupů pro každou stránku. Ukázka tohoto grafu je vidět na obrázku 5-7. Tento graf poskytuje přehled o navštěvovaných stránkách a lze z něj snadno zjistit, které stránky byly navštěvovány nejčastěji, případně které nejméně.

Následuje tabulka cest, jejíž pop-up menu je tvořeno stejně jako u předchozí tabulky 3 položkami. Stejný zůstává i účel těchto položek. Tedy první položka provádí export dat z tabulky do formátu pdf, druhá zobrazuje okno s detailními informacemi o zvolené cestě a poslední položka zobrazuje graf návštěvnosti stránek. Rozdíl spočívá pouze v možných rozdílných datech v jednotlivých tabulkách, protože jak již bylo vysvětleno, v rámci identifikace cest se aplikace snaží dokončit mapování pohybu uživatele po zkoumaném webu. A z tohoto důvodu se mohou oba grafy mírně lišit.

Předposlední tabulkou je tabulka transakce, které co do počtu položek pop-up menu je nejchudší. Její pop-up menu obsahuje pouze dvě položky. První položka slouží opět pro export dat z tabulky do dokumentu formátu pdf. Druhá položka zobrazuje detail o zvolené transakci. Podobně jako u předchozích dvou tabulek.

Poslední tabulkou je tabulky asociačních pravidel. Její pop-up menu obsahuje 3 položky. První položka, stejně jako ve všech ostatních případech, slouží k exportu dat z tabulky do dokumentu pdf. Druhá položka vytváří okno s grafem zobrazují statistiku přístupů robotů na web. Graf pro každý den ze zkoumaného logu zobrazuje počet robotů, kteří v daný den vstoupili na web. Poslední položka pop-up menu zobrazuje také okno s grafem, který podobně jako předchozí graf zobrazuje statistiku

přístupů robotů na web. Oproti předchozímu grafu se ale jedná o odlišnou statistiku. Místo počtu přístupů robotů na web každý den zobrazuje, kolikrát konkrétní robot vstoupil na stránky. Graf tedy zobrazuje jména všech robotů, kteří byli v logu rozpoznání, a ke každému jménu počet přístupů.



Obrázek 5-7 Statistika navštívených stránek

6 Experimenty

Tato kapitola je věnována interpretaci výsledků získaných při testování aplikace, zkoumání chování aplikace a celkově vývoji aplikace. V rámci kapitoly budou vysvětleny významy získaných výsledků, a jak se tyto výsledky mění při změně parametrů aplikace. Čtenář tak získá přehled o tom, jak jednotlivé nastavení mohou ovlivnit získané výsledky. Při vývoji aplikace bylo použito více webových logů, a to hlavně z důvodu pokrytí co možná největšího množství situací, které mohou nastat. Interpretovány budou převážně výsledky získané z jednoho logu, ale pro lepší vysvětlení, jak některé nastavení ovlivňují získané výsledky, budou využity i další logy.

Výsledkem provádění aplikace je množina asociačních pravidel. Nejčastější asociační pravidla jsou tvořena dvojicemi. V závislosti na zkoumaném logu je však možné získat i asociační pravidla tvořená obecně n-ticemi. V aplikaci je nalezení asociačních pravidel ovlivněno několika nastaveními. Nastavení, které nejvíce ovlivňují nalezení asociačních pravidel, jsou zcela jistě nastavení parametrů support a confidence. Tyto parametry jsou v základním nastavení aplikace nastaveny na hodnoty support = 10%, confidence = 40%. Výchozí hodnoty těchto parametrů byly určeny na základě pozorování během vývoje aplikace. Hodnota parametru support byla zvolena 10% převážně z důvodu, že v případě nastavení vyšší hodnoty se v některých případech stávalo, že aplikace poskytla příliš málo asociačních pravidel, nebo dokonce žádné. Je poměrně intuitivní, že vzhledem k tomu, že tento parametr udává, v kolika procentech případů se vyskytují všechny položky daného pravidla společně v rámci jedné transakce, není možné, aby byla hodnota daného parametru vysoká. Pravděpodobnost nalezení asociačních pravidel složených z více než dvou položek se zvyšuje se snížením hodnoty tohoto parametru. Výjimku mohou tvořit webové logy, které obsahují přístupy na malou množinu stránek. Druhým parametrem je confidence, který udává procentu případů, ve kterých, jestliže se vyskytuje první část pravidla, tak se také vyskytuje druhá část pravidla. Nastavení hodnoty tohoto parametru je dle mínění autora složitější než u předchozího parametru, a také je do značné míry subjektivní. V některých případech může totiž vysoká hodnota tohoto parametru u pravidla naznačovat, že se jedná o logický důsledek organizace odkazů na zkoumaném Webu, v jiném případě naopak o zjištění zajímavé a neočekávané skutečnosti. Posouzení, o který z uvedených případů se jedná, je nutné ponechat konkrétní situaci, kdy je možné vzít v úvahu organizaci zkoumaného Webu. Z uvedených důvodů je hodnota nastavena na 40%, která umožňuje zachycení nejrůznějších případů, a to ať už s vyšší hodnotou tohoto parametru, nebo nižší.

Nyní si představíme výsledek zpracování webového logu pizzerie, v základním nastavení, který byl použit při vývoji aplikace. Výsledkem jsou 3 asociační pravidla tvořená dvojicemi a jedno pravidlo tvořené trojicí. Jedná se o následující asociační pravidla, index -> nabídka, menu -> nabídka, menu -> index a menu -> index -> nabídka. V grafické podobě je možné tyto pravidla vidět na Obrázku 7-1, který je součástí přílohy. První pravidlo říká, že v 42% jde návštěvník z úvodní stránky

na stránku s nabídkou jídel. A dále že pokud návštěvník vstoupí na úvodní stránku, tak v téměř 60% případů, poté také vstoupí na stránku s nabídkou jídel. Druhé pravidlo poté obdobně poskytuje informaci, že ve 21% případů přechází návštěvník ze stránky s aktuálním menu na stránku s kompletní nabídkou jídel. Poslední, tedy čtvrté pravidlo, je složeno z trojice stránek a udává, že v 19% případů využívá návštěvník pro přechod ze stránky s týdenním menu na stránku s nabídkou jídel úvodní stránku. Toto pravidlo by tak např. mohlo znamenat, že neexistuje přímý odkaz ze stránky s nabídkou týdenního menu na stránku s nabídkou jídel, nebo odkaz existuje, ale není vhodně umístěný a uživatelé jej snadno přehlédnou. Pokud změníme hodnoty uvedených parametrů, získáme nové asociační pravidla. Změníme tedy u stejného webového logu parametry na support = 5 % a confidence = 20 %. Po vyhledání pravidel s upravenými parametry jsme nyní získali 10 asociačních pravidel. Z těchto pravidel je 6 pravidel nových a zbylé 4 jsou stávající. Nové pravidla jsou následující, menu -> kontakt, index -> kontakt, nabídka -> kontakt, index -> nabídka -> kontakt, menu -> nabídka -> kontakt a menu -> index -> nabídka -> kontakt (viz. Obrázek 7-2). Lze si povšimnout, že některá pravidla rozšiřují pravidla nalezená se základním nastavením. Jako příklad lze uvézt pravidla index -> nabídka a nově nalezené pravidlo index -> nabídka -> kontakt. Toto nové pravidlo lze snadno chápat tak, že návštěvník přijde na web, přejde na nabídku jídel a po vybrání jídla hledá kontakt pro objednání jídla. Je také poměrně intuitivní, že parametry tohoto nového pravidla musí být nižší, neboť stálý zákazníci již např. mohou mít kontakt uložen v mobilu a nemusí tak kontakt pro objednání hledat. Dalším příkladem rozšíření může být pravidlo menu -> index -> nabídka a nové pravidlo menu -> index -> nabídka -> kontakt. Toto pravidlo lze chápat obdobně, jako již vysvětlené. Dále se v tomto případě ukázalo, že jako významnější se prokázala změna parametru confidence. Pokud by tento parametr nebyl změněn, nedošlo by k nalezení všech 6 nových pravidel. Zatímco pokud by zůstal nezměněn parametr support, došlo by k nalezení 2 nových pravidel a zbylé 4 by nalezeny nebyly. V tomto konkrétním případě se tedy změna parametru confidence ukázala jako významnější, ovšem je nutné zdůraznit, že tato skutečnost není pravidlem. Míra, jakou změna uvedených parametrů ovlivní nalezení nových asociačních pravidel, je různá u každého zkoumaného logu. Dále si také můžeme povšimnout, že z 10 pravidel je 6 pravidel tvořeno dvojicí, 3 trojicí a jedno pravidlo je dokonce tvořeno čtveřicí stránek. Lze vypozorovat, že většina pravidel tvořených větším počtem stránek má nižší hodnotu parametrů, zejména parametru support. Obecně lze tedy říci, že čím nižší je hodnota parametrů support a confidence, tím je nalezeno více složitějších pravidel. Nová pravidla můžeme interpretovat jako v předchozím případě. Pokud bychom tedy vzali nejsložitější pravidlo, tak toto pravidlo znamená, že v 6 % návštěvník přešel ze stránky aktuálního menu na úvodní stránku, následně na nabídku jídel a poté na stránku s kontakty. Je možné usoudit, že tito návštěvníci si vybírali jídlo nejprve v aktuálním menu, následně v kompletní nabídce a poté hledali kontakt pro objednání jídla. Jak bylo uvedeno, parametry support a confidence ovlivňují nalezení asociačních pravidel v největším rozsahu. Nejedná se však o jedinou možnost nastavení, která ve svém důsledku ovlivňuje vyhledání asociačních pravidel. Další z možností je změnit způsob určení

transakcí a případně sezení. Jak bylo popsáno v předchozích kapitolách, aplikace poskytuje 2 metody určení transakcí a 3 metody určení sezení. Ponecháme tedy stejné hodnoty parametrů support a confidence, jako v předchozím příkladu, a změníme způsob identifikace transakcí na metodu maximal forward reference a metodu identifikace sezení na maximální odstup mezi dotazy. Výsledkem hledání asociačních pravidel jsou stejná pravidla, jako v předchozím hledání. Nalezeno tedy bylo 10 pravidel. Podíváme se tedy nyní detailněji na výsledná pravidla a jejich parametry. V prvním případě má pravidlo index -> nabídka hodnoty parametrů support a confidence rovny 42,18% a 42,45%. V druhém případě u stejného pravidla jsou hodnoty rovny 42,45% a 60,2%. Lze si povšimnout malých rozdílů v hodnotách parametrů. V uvedeném případě se sice jedná pouze o desetiny procent, nicméně i desetiny procent mohou mít vliv na skutečnost, zda pravidlo bude nalezeno či nikoliv. Představme si případ, kdy by hodnota parametru confidence byla nastavena na 60 %. V takovém případě by uvedené pravidlo nebylo v prvním případě nalezeno, a naopak ve druhém případě by pravidlo nalezeno bylo. Podobným způsobem, jako u uvedeného pravidla, se liší i všech zbylých 9 pravidel. V převážné většině pravidel se jedná o desetiny procent. V průběhu vývoje a testování aplikace však bylo zjištěno, že se tyto hodnoty mohou lišit až v řádu procent. Vše opět závisí na konkrétním webovém logu. Ať už se však hodnoty parametrů pravidel liší o desetiny procent, či celá procenta, může tato skutečnost vézt k situaci, kdy pravidla na hraně nastavených hodnot budou, nebo nebudou nalezena. Poslední z možností jak ovlivnit vyhledání asociačních pravidel, je přidání přípon souborů, které se mají do analýzy zahrnout. Jak bylo dříve vysvětleno, v základním nastavení uvažuje aplikace pouze webové stránky a ostatní typy souborů, jako mohou být videa, dokumenty, atd. při analýze vyřazuje ve fázi čištění dat. V některých případech ovšem mohou např. dokumenty tvořit podstatnou část analyzovaného Webu, a je proto příhodné je do analýzy zahrnout. Proto aplikace umožňuje zvolit přípony souborů, které chceme v analýze zohlednit. Tato možnost nastavení představuje podobně jako nastavení parametrů support a confidence významný prvek při vyhledávání asociačních pravidel. Proveďme tedy další vyhledání asociačních pravidel. Tentokrát s nastavením parametrů support = 5 %, confidence = 5 % a zahrnutím formátu pdf do procesu hledání asociačních pravidel. Jako výsledek hledání jsme získali 19 pravidel. Z těchto 19 pravidel je 10 pravidel shodných s výsledky v předchozích případech. Ze zbylých 9 pravidel je 8 pouhým důsledkem snížení hodnoty parametru confidence na 5 % a poslední pravidlo je důsledkem zahrnutí formátu pdf analýzy. Toto pravidlo má následující tvar, index -> menu.pdf (viz. Obrázek 7-3). V důsledku zahrnutí formátu pdf do analýzy webového logu jsme tedy získali jedno nové pravidlo. Toto nové pravidlo vyjadřuje skutečnost, že v 6 % případů se v transakci současně vyskytuje úvodní stránka a soubor menu.pdf, který obsahuje aktuální týdenní menu. A dále také, že pokud se vyskytne úvodní stránka, tak v 8 % případů se také vyskytne soubor s aktuálním menu. Nicméně zahrnutí formátu pdf do analýzy nemá za následek pouze nalezení nového pravidla. V důsledku zahrnutí formátu pdf do analýzy dochází i ke změně hodnot parametrů u nalezených pravidel. Vezměme si opět např. pravidlo index -> nabídka. V úplně prvním případě hledání asociačních pravidel, uvedeném dříve, měly parametry tohoto

pravidla hodnoty 42,18 % a 59,93 %. V aktuálním případě jsou hodnoty parametrů stejného pravidla rovny hodnotám 41,01 % a 58,17 %. Tato skutečnost je způsobena faktem, že v důsledku zahrnutí formátu pdf do analýzy se zvýšil počet zkoumaných položek. A tedy i hodnoty parametrů jednotlivých pravidel se zákonitě musí změnit. Skutečnost, o kolik se budou hodnoty parametrů pravidel lišit, závisí na faktu, o kolik se zvýšil počet položek ke zkoumání. Některé hodnoty se mohou lišit v řádech procent. Na základě nově nalezeného pravidla a pravidel získaných a popsaných dříve, můžeme vyvodit fakt, že uživatelé zkoumaného Webu dávají přednost online zobrazení aktuálního menu před možností stáhnout si toto menu jako pdf soubor. Další příklad rozdílů při vyhledávání asociačních pravidel s různým nastavením ilustrují obrázek 7-4 a obrázek 7-5. Tyto obrázky představují situaci, kdy v případě obrázku 7-4 se vyhledávají pouze webové stránky a v případě obrázku 7-5 se navíc vyhledávají soubory ve formátu doc. Na uvedených obrázcích vpravo nahoře můžeme vidět, že v prvním případě je počet nalezených asociačních pravidel roven 17 a v druhém 32. Na základě uvedených informací je možné prohlásit, že volba souborů, které do analýzy zahrneme, je velmi důležitá z hlediska počtu nalezených pravidel a hodnot parametrů jednotlivých pravidel. Je ovšem také nutné uvést, že nastavení typů souborů se projeví pouze tehdy, pokud zkoumaný webový log obsahuje soubory zvolených typů. V opačném případě se toto nastavení žádným způsobem neprojeví.

Časová složitost nalezení asociačních pravidel, a tedy celé aplikace, je dána složitostí jednotlivých kroků předzpracování dat a složitostí algoritmu Apriori. Prvním krokem je načtení dat ze souboru, jehož časová složitost je lineární. Důvodem je skutečnost, že data jsou načítána v cyklu, který se provede n-krát, kde n je počet řádků logu. Jako další se v procesu předzpracování dat provádí čištění dat. Složitost tohoto kroku je taktéž lineární, jelikož se v cyklu prochází záznamy načtené v předchozím kroku. Krok čištění dat je však velmi významný z pohledu dalších kroků fáze předzpracování dat. A to z důvodu redukce počtu záznamů, které se budou v dalších krocích zpracovávat. Čištění dat má tedy potenciál velkou měrou snížit dobu potřebnou pro vykonání následujících kroků předzpracování dat. Následujícím krokem je identifikace uživatelů, která je prováděna v cyklu, jež je vykonán m-krát, kde m je počet záznamů získaných čištěním dat. Časová složitost identifikace uživatelů je tedy opět lineární. Dalším krokem je získání sezení. Tento krok probíhá ve dvou vnořených cyklech, první je proveden n-krát a druhý je proveden m-krát, kde n je počet nalezených uživatelů a m je počet záznamů uživatele ve zkoumaném logu. Časová složitost je tedy O (n*m). Při zamyšlení se nad uvedenou složitostí ovšem zjistíme, že n*m udává stejnou hodnotu, jakou jsme získali po provedení kroku čištění dat, a tedy že součin uvedených cyklů je stejný jako v případě identifikace uživatelů. Podobný postup je použit i pro nalezení cest a transakcí. Pouze s tím rozdílem, že při identifikaci transakcí může být cyklus obecně vykonán vícekrát, než je počet záznamů získaných po čištění dat. To je způsobeno faktem, že při hledání cest jsou doplňovány záznamy, které v logu nejsou evidovány např. z důvodu použití tlačítka zpět. Posledním krokem je nalezení samotných asociačních pravidel, které je realizováno algoritmem Apriori. Dle [5] je

teoretická časová složitost zmíněného algoritmu $2^{O(n)}$. Nicméně dolovací algoritmus využívá řídkosti dat a minimální hodnoty parametru support, díky čemuž je dolování uskutečnitelné a efektivní. Dle výše uvedeného je časová složitost nejhůře kvadratická.

7 Závěr

Cílem práce bylo seznámit se s problematikou dolování na Webu se zaměřením na dolování z užití Webu, navržení koncepce aplikace schopné provádět předzpracování dat z webového logu a nalezení asociačních pravidel a dle navržené koncepce aplikaci následně implementovat. Posledním cílem je ověřit funkčnost aplikace a provedení experimentů. Kapitoly 2 a 3 se zabývají definicí důležitých pojmů související s řešenou problematikou. Při vytváření výše uvedených kapitol a čtení zdrojů uvedených v seznamu literatury jsem se seznámil nejen s problematikou dolování na Webu, ale i s problematikou dolování dat obecně. Získané znalosti považuji za velice přínosné, neboť do značné míry ovlivnily můj pohled na dolování dat obecně, a také na analýzu užívání Webu. Druhým cílem práce bylo vytvoření koncepce aplikace. Vytvořením návrhu aplikace se zabývá kapitola 4. V rámci návrhu byl vytvořen konceptuální diagram tříd (viz. Obrázek 4-1) a diagram případů užití (viz. Obrázek 4-2). Třetím cílem bylo na základě vytvořeného návrhu aplikaci implementovat. Implementací aplikace se zabývá kapitola č. 5. V kapitole je popsána implementace nejdůležitějších funkcí aplikace, která je doplněna ukázkami obrázků aplikace. Kapitola č. 6 se zabývá experimenty s vytvořenou aplikací a intepretací získaných výsledků, čímž byl splněn další cíl této práce.

V rámci práce byla vytvořena funkční aplikace, která umožňuje zpracovat webový log ve formátu NCSA Combined a získat asociační pravidla udávající chování návštěvníků zkoumaného Webu. V současnosti umožňuje aplikace zpracovat pouze log ve formátu NCSA Combined, a proto se jako primární možné pokračování projektu jeví možnost doplnit aplikaci o schopnost zpracovat i další formáty webového logu. Tato schopnost je v aplikaci rovněž naznačena možností volby formátu webového logu, která je ovšem v současné verzi aplikace nefunkční. A proto je formát webového logu v současnosti neměnný. Další možností je rozšíření aplikace o automatickou aktualizaci databáze známých robotů, která by vylepšila schopnost aplikace odfiltrovat nežádoucí záznamy ve webovém logu. Pro rozšíření funkčnosti aplikace se ovšem jako klíčové jeví již uvedené rozšíření aplikace a možnost práce i s ostatními formáty webových logů.

Literatura

[1] U. Fayyad, G. Piatetsky-Shapiro a P. Smyth, „From Data Mining to Knowledge Discovery in Databases," *AI MAGAZINE*, sv. 17, č. 3, pp. 37-54, 1996.

[2] S. Chakrabarti, B. E. Dom, S. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson a J. Kleinberg, „Mining the Link Structure of the World Wide Web," *IEEE Komputer*, sv. 32, č. 8, pp. 60-67, 1999.

[3] J. Srivastava, R. Cooley, M. Deshpande a P.-N. Tan, „Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data," *SIGKDD Explorations*, sv. 1, č. 2, pp. 12-23, Leden 2000.

[4] B. Berendt, B. Mobasher, M. Nakagawa a M. Spiliopoulou, „The Impact of Site Structure and User Environment on Session Reconstruction in Web Usage Analysis," *WEBKDD*, sv. 2703, pp. 159-179, 2002.

[5] B. Liu, Web Data Mining Exploring Hyperlinks, Contents, and Usage Data, Springer Berlin Heidelberg, 2011.

[6] G. Paliouras, C. Papatheodorou, V. Karkaletsis a C. D. Spyropoulos, „Discovering User Communities on the Internet Using Unsupervised Machine Learning Techniques," *Interacting with Computers*, pp. 761-791, 2002.

[7] R. Agrawal, T. Imieliński a A. Swami, „Mining association rules between sets of items in large databases," *ACM SIGMOD Record*, sv. 22, č. 2, pp. 207-216, Červen 1993.

[8] M. Dimitrijević a Z. Bošnjak, „Discovering Interesting Association Rules in the Web Log Usage Data," *Interdisciplinary Journal of Information, Knowledge, and Management*, sv. 5, pp. 191-207, 2010.

[9] H. Wang, C. Yang a H. Zeng, „Design and Implementation of a Web Usage Mining Model Based On Fpgrowth and Prefixspan," *Communications of the IIMA*, sv. 6, č. 2, pp. 71-86, 2006.

[10] J. Mallat, „Analýza UA :: user-agent-string.info," [Online]. Available: http://user-agent-string.info/. [Přístup získán Březen 2013].

[11] A. Staeding, „List of User-Agents (Spiders, Robots, Crawler, Browser)," [Online]. Available: http://www.user-agents.org/. [Přístup získán Únor 2013].

[12] Z. Markov a D. T. Larose, Data Mining The Web : Uncovering Patterns in Web Content, Structure and Usage, Hoboken, New Jersey: Wiley, 2007.

[13] D. T. Larose, Discovering Knowledge in Data : An Introduction to Data Mining, Hoboken, New Jersey: Wiley, 2005.

[14] R. Cooley, *Web Usage Mining: Discovery and Application of Interesting Patterns from Web data*, PhD thesis, Dept. of Computer Science, University of Minnesota, 2000.

[15] M. Baglioni, U. Ferrara, A. Romei, S. Ruggieri a F. Turini, „Preprocessing and Mining Web Log Data for Web Personalization," *Advances in Artificial Intelligence*, sv. 2829, pp. 237-249, 2003.

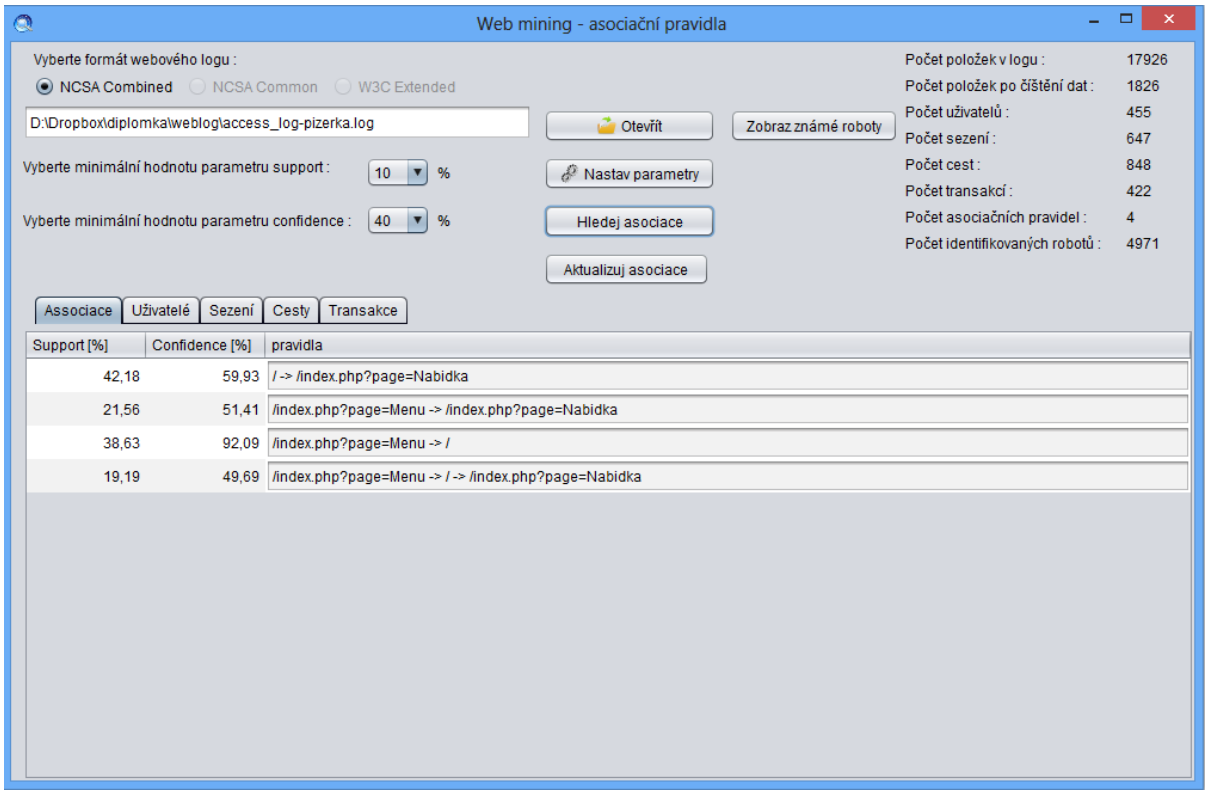
[16] B. Mobasher, R. Cooley a J. Srivastava, „Automatic Personalization Based on Web Usage Mining," *Communications of the ACM*, sv. 43, č. 8, pp. 142-151, Srpen 2000.

[17] R. Cooley, B. Mobasher a J. Srivastava, „Data Preparation for Mining World Wide Web Browsing Patterns," *Knowledge and Information Systems*, sv. 1, pp. 5-52, 1999.

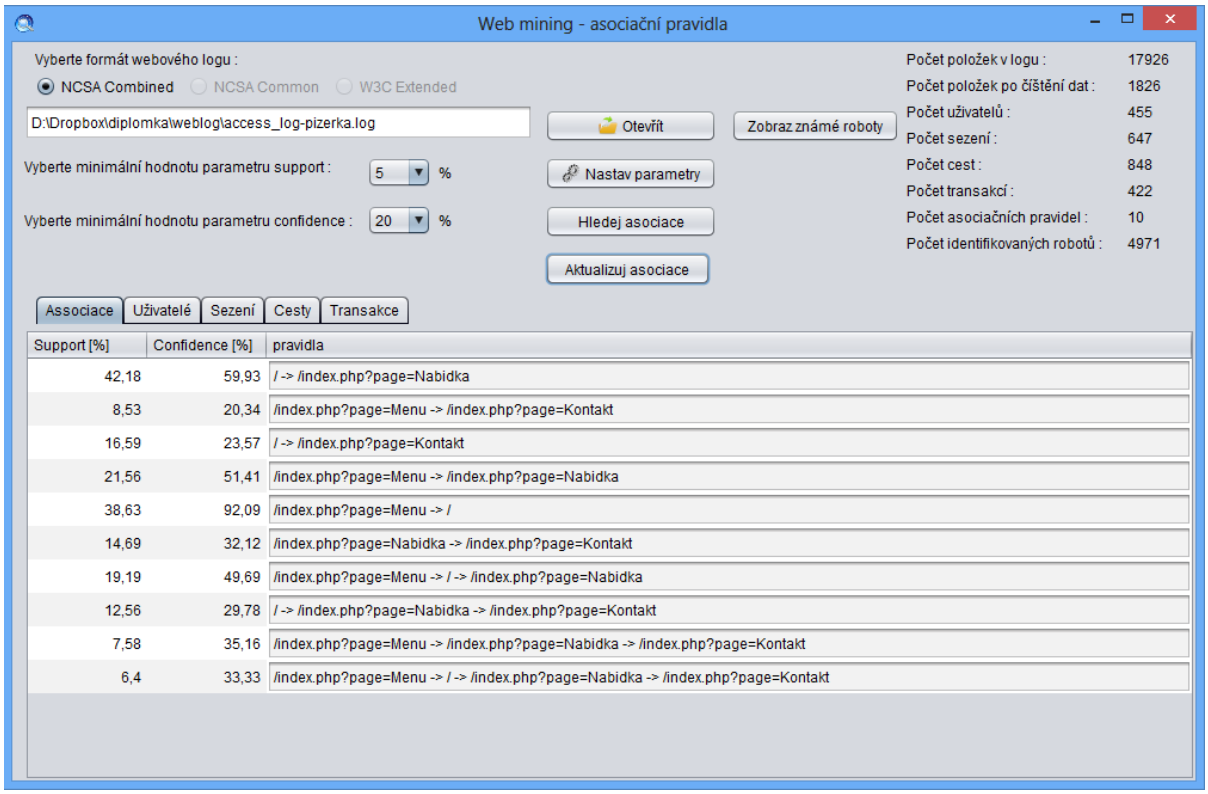
[18] S. K. Pani, L. Panigrahy, V. H. Sankar, B. K. Ratha, A. K. Mandal a S. K. Pandhi, „Web Usage Mining : A Survey on Pattern Extraction from Web Logs," *International Journal of Instrumentation, Control & Automation*, pp. 15-23, 2011.

[19] Y. H. Cho, J. K. Kim a S. H. Kim, „A Personalized Recommender System Based on Web Usage Mining," *Expert Systems with Applications*, pp. 329-342, 2002.

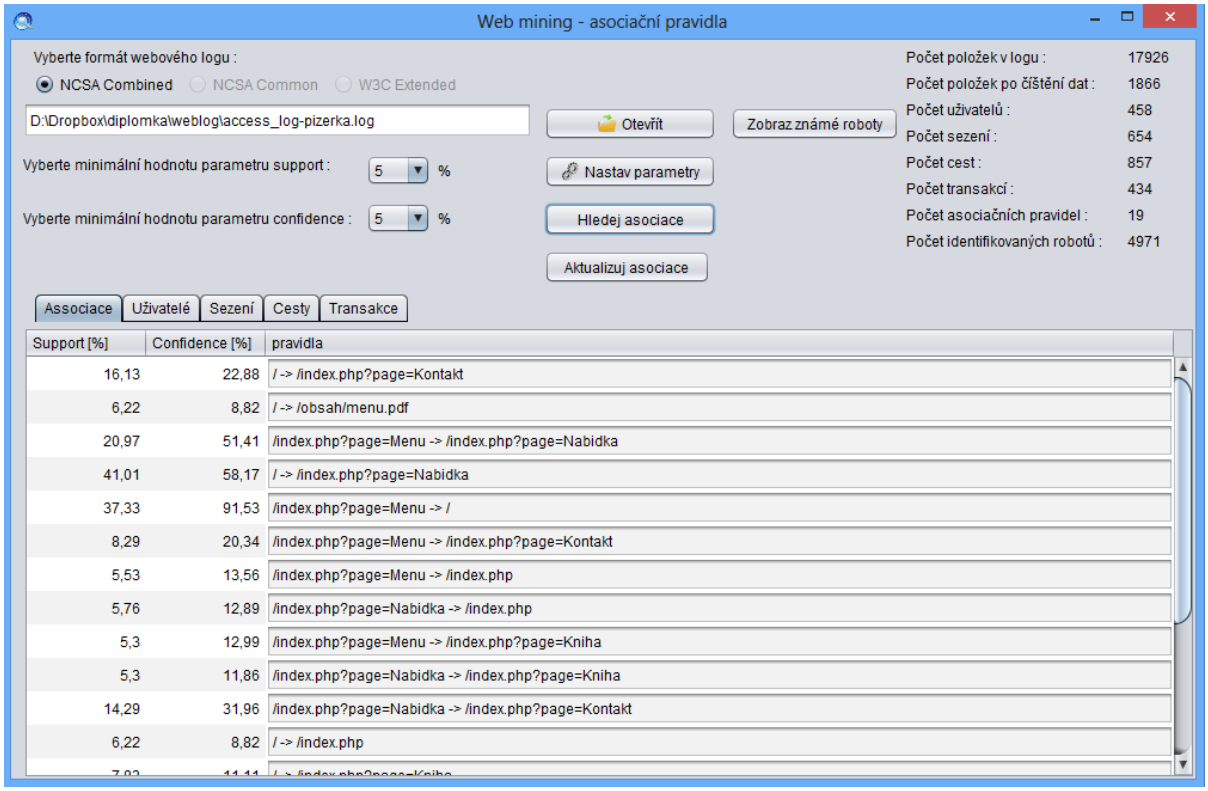
Seznam příloh



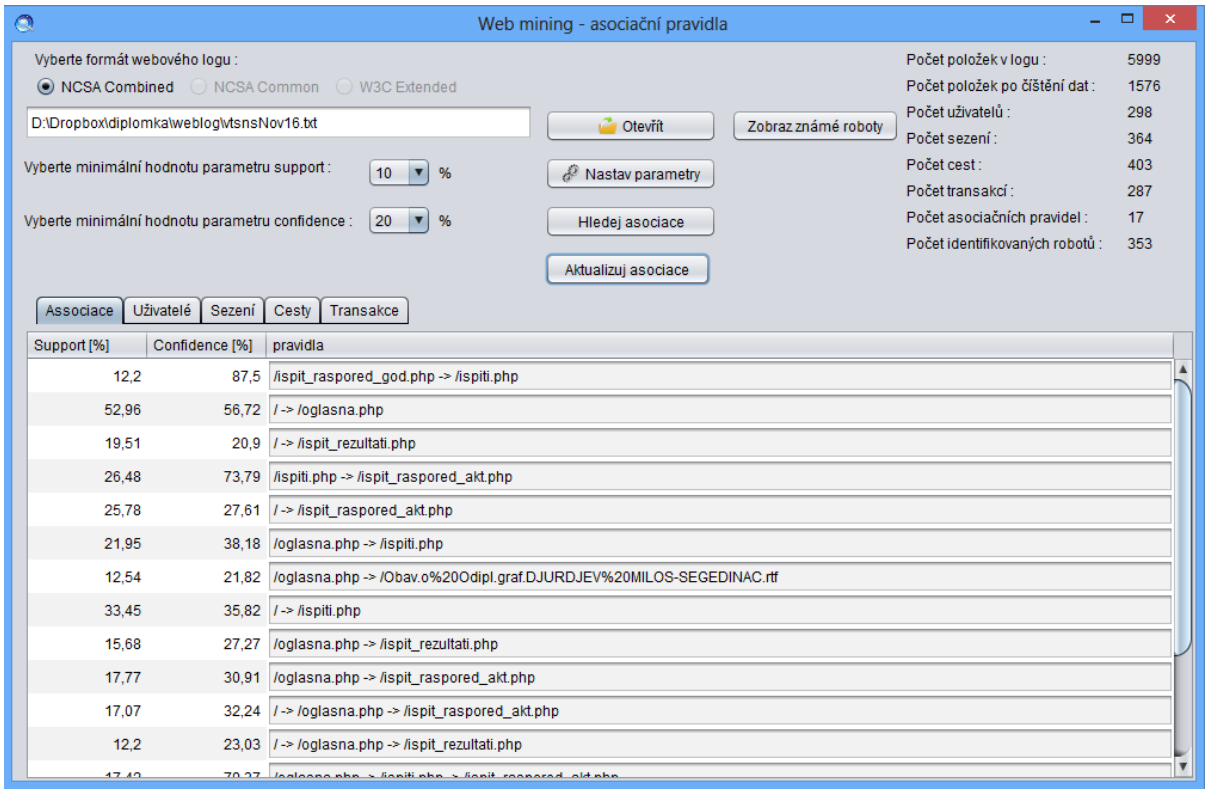
Obrázek 7-1 Asociační pravidla získaná v základním nastavení



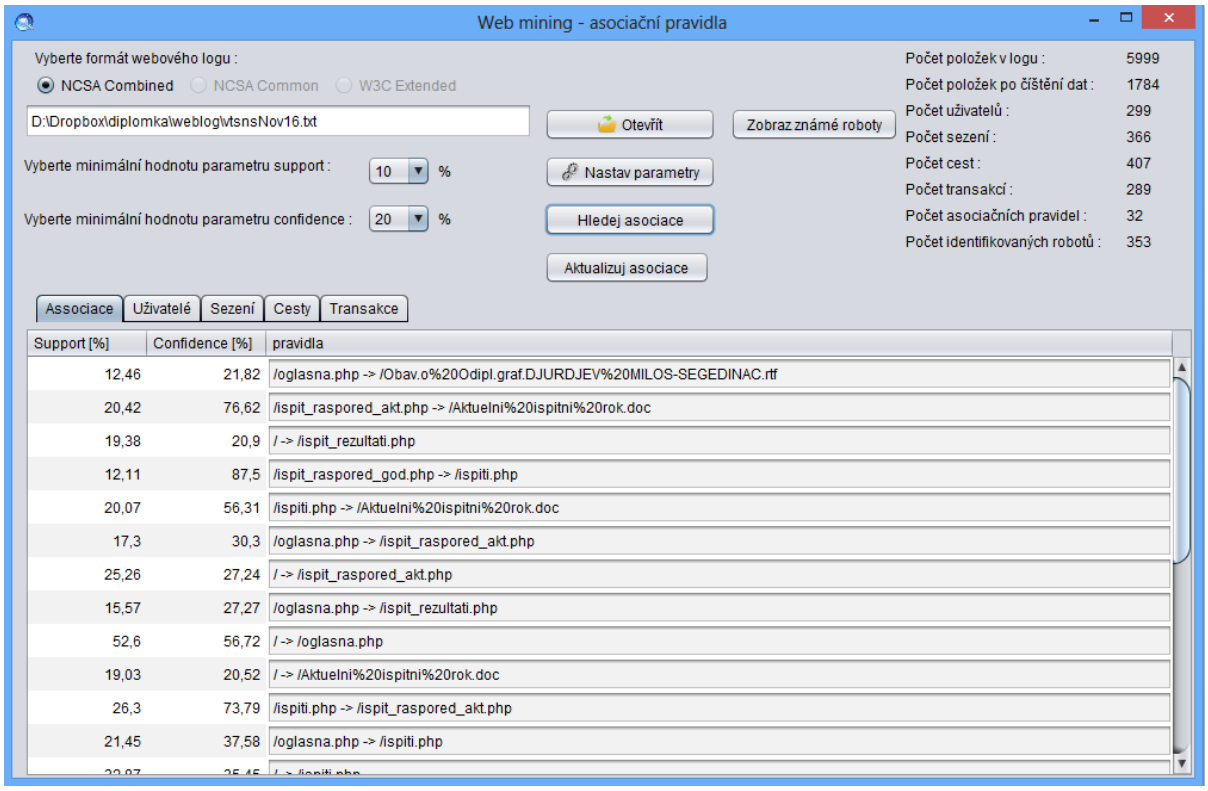
Obrázek 7-2 Asociační pravidla získaná upravením parametrů



Obrázek 7-3 Asociační pravidla se zahrnutím formátu pdf



Obrázek 7-4 Asociační pravidla bez souborů .doc



Obrázek 7-5 Asociační pravidla se soubory .doc